DIVA.EXCHANGE

DNS for I2P: a Distributed Network without Central Authority

How Students Tried to Create a DNS for an Overlay Network without a Central Authority

Author: Konrad Bächler
Twitter: @DigitalValueX, Web: https://diva.exchange

About diva.exchange

- Non profit association, open to everyone
- A loose bunch of Devs & Researchers - spread all over the world
- «DIVA - Free Banking Technology for Everyone» means: handle all kind of Digital Values under your own control and responsibility and apply your very own philosophy of privacy without being nudged by others
- No centralized business model (pointless); no token/coin.

Agenda

- Concept of the I2P network
- Motivation: Why I2P needs a DNS
- Setup: tools, blockchain and an I2P test network
- Byzantine Fault Tolerance as consensus algorithm
- Design and usage of the DNS API for I2P
- Discussion: feedback from DNS devs

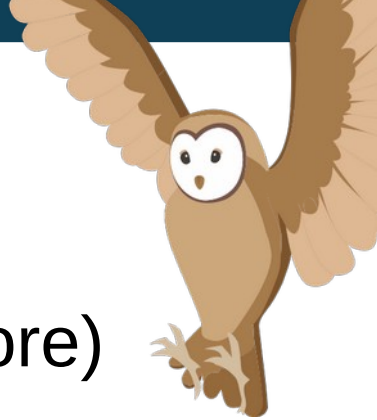
Hello I2P Network

- A few basic facts (some are simplified - educational reasons):
  - I2P is an overlay network (misleading name «darknet» is just used by dubious media desperately in need for clicks)
  - It's a peer-to-peer network where every node in the network acts as a router
  - I2P itself has no storage capabilities – it is a transport layer
  - Messages travelling through the network are multiple times encrypted (like a garlic: it has multiple layers) – call it «Confidentiality feature»
  - Messages hop over several routers within the network to their final destination (using «tunnels») – call it «Anonymity feature»
- In a nutshell: I2P = confidential & anonymous message transport

Concept: I2P-b32 Adresses

- Remember: the I2P network provides anonymity
- Example:
  auoqibfnyujhcht4v3nzahpqztwlyomesfywltuls5bqqi3nd3ka = diva.i2p
- Destinations within the network («peers») are identified by public keys
- A Public Key of a destination can be transformed to a b32 address. I2P-b32 address = the base32 encoded sha26-hash of the public key
- There exists no algorithm to resolve a name (like «diva») to it's b32 address – it's only a local lookup in a key/value store

Concept (2)



- I2P has only a local addressbook (a key/value store)
- Users can build their own addressbook or download it from a «trusted» source (which is a joke by itself in an anonymous and trustless network)
- Let's study a very trivial use case of DNS: mapping destinations to shorter and simpler names

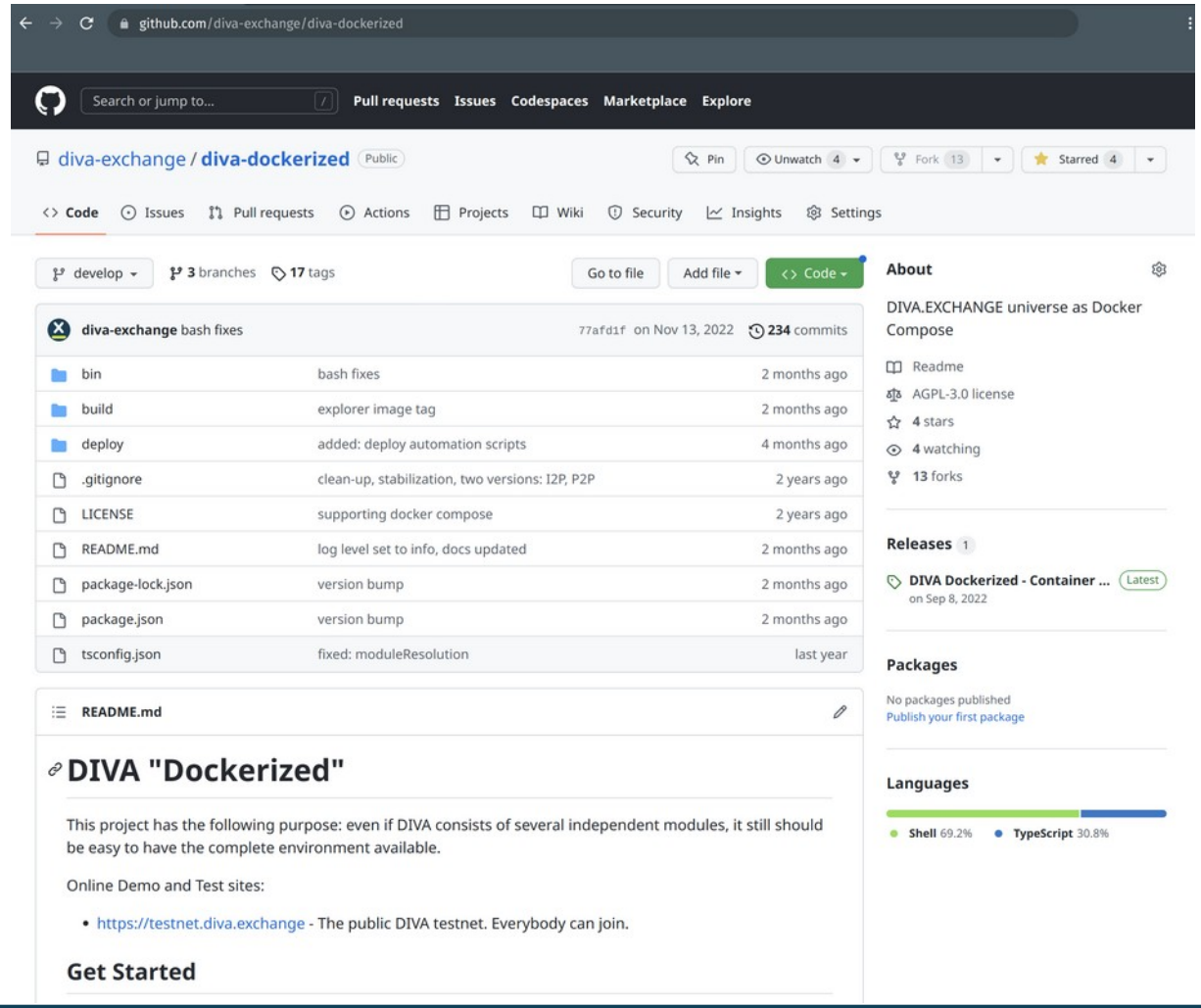
Motivation: Why?

- I2P shall be a fully distributed network – without central authority
- Today, the I2P network has some «jump services and registries» which act as kind-of-DNS. This is obviously a central component and hence not compatible with the idea of true distribution (not saying that there are bad actors, it's just not compatible).
- True distribution = distributed ledger = blockchain
- Students of the University of Applied Science and Arts, Lucerne, Switzerland, were highly motivated to create a prototype

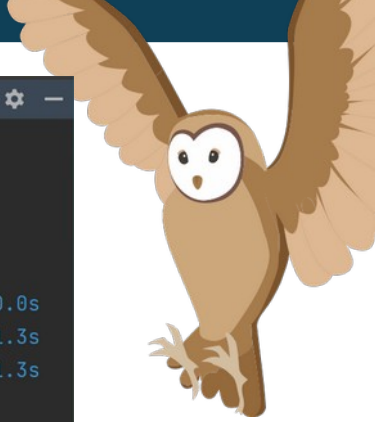
Setup: DIVA blockchain

- The students used https://github.com/diva-exchange/diva-dockerized to get their test environment up and running
- Divachain (the blockchain) uses Democratic Byzantine Fault Tolerance (D-BFT) as consensus. D-BFT got mainly developed by diva.exchange since 2020 but also at other universities in other corners of the world, like in Sydney (AUS) or Lausanne (CHE).
- From a blockchain-development perspective D-BFT, in combination with Proof-of-Stake, is one correct solution to the real-world problems given in fully distributed and trustless systems (like reliability, independence, performance, resource-efficiency)

Setup


github.com/diva-exchange/diva-dockerized
Search or jump to...
Pull requests
Issues
Codespaces
Marketplace
Explore
diva-exchange / diva-dockerized
Public
Pin
Unwatch 4
Fork 13
Starred 4
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
develop
3 branches
17 tags
Go to file
Add file
Code
diva-exchange bash fixes
77afd1f on Nov 13, 2022
234 commits
bin bash fixes 2 months ago
build explorer image tag 2 months ago
deploy added: deploy automation scripts 4 months ago
.gitignore clean-up, stabilization, two versions: I2P, P2P 2 years ago
LICENSE supporting docker compose 2 years ago
README.md log level set to info, docs updated 2 months ago
package-lock.json version bump 2 months ago
package.json version bump 2 months ago
tsconfig.json fixed: moduleResolution last year
README.md
DIVA "Dockerized"
This project has the following purpose: even if DIVA consists of several independent modules, it still should be easy to have the complete environment available.
Online Demo and Test sites:
https://testnet.diva.exchange - The public DIVA testnet. Everybody can join.
Get Started
About
DIVA.EXCHANGE universe as Docker Compose
Readme
AGPL-3.0 license
4 stars
4 watching
13 forks
Releases 1
DIVA Dockerized - Container ... Latest
on Sep 8, 2022
Packages
No packages published
Publish your first package
Languages
Shell 69.2%
TypeScript 30.8%

Setup

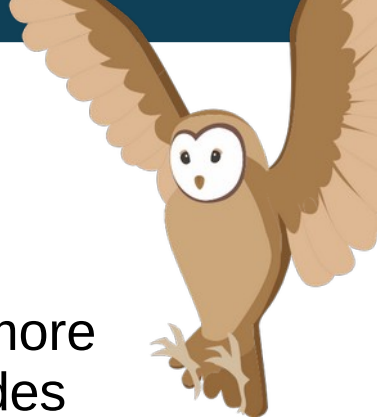


```
Terminal:  Local
$
$ DIVA_TESTNET=1 bin/build.sh

⇒  Creating Genesis Block using I2P…
[+] Running 3/3
 ⠿ genesis-i2p-udp Skipped - Image is already being pulled by genesis-i2p-http   0.0s
 ⠿ genesis-i2p-chain Pulled                                                     1.3s
 ⠿ genesis-i2p-http Pulled                                                      1.3s
[+] Running 4/4
 ⠿ Network network.genesis-i2p  Created                                         0.1s
 ⠿ Container genesis-i2p-udp    Started                                         0.4s
 ⠿ Container genesis-i2p-http   Started                                         0.4s
 ⠿ Container genesis-i2p-chain  Started                                         0.7s

⇒  Waiting for key generation…
[+] Running 4/4
 ⠿ Container genesis-i2p-chain  Removed                                         0.2s
 ⠿ Container genesis-i2p-udp    Removed                                        10.4s
 ⠿ Container genesis-i2p-http   Removed                                        10.3s
 ⠿ Network network.genesis-i2p  Removed                                         0.4s

[ok]  Genesis Block successfully created

⇒  Creating diva.yml file…

[ok]  Created diva.yml file
$
```

Setup

- DIVA_TESTNET=1 bin/build.sh
- DIVA_TESTNET=1 bin/start.sh
- Result:
  - A bundle of docker container, forming a local node connected via I2P to the fully distributed DIVA blockchain
  - This environment is ready to work as a fully distributed and truly private storage layer for arbitrary small data
- Purge all your local data like this:
  - DIVA_TESTNET=1 bin/purge.sh

Democratic BFT



- Byzantine Fault Tolerance in a nutshell: a distributed system with more than two nodes is reliably working as long as «faulty plus one» nodes are correctly working.
- The Binary Byzantine Consensus: assume correct processes propose a binary value {0, 1}. Now, a reliable decision in a fully distributed system is possible, if:
  - Every correct node takes a decision
  - All correct nodes take the same decision
- Now, a blockchain faces a «Set problem» (a block is a set of transactions) – this is an extension of the Binary Byzantine Consensus

DNS for I2P: Specs

- Students had to follow these specs:
  - Write to the chain, a key/value-pair (kind of a «DNS A» record)
  - Read from the chain, whereas the search string is the name (like «diva.i2p») and it shall return the I2P-b32-address
  - It had to be implemented as local HTTP API and delivered as a docker container
- Full specs, in German language, here:

  https://gist.github.com/diva-exchange/aa6b1adbfefe909cd3ea07ac3cdfc322

DNS for I2P: Solutions

- Multiple prototype implementations in several languages available:
  - https://github.com/diva-exchange/i2p-dns
- Reading and writing to the fully distributed DNS is working
- «Decisions» couldnot be implemented, since the divachain release got delayed (D-BFT issues in combination with Proof-of-Stake)

Examples

- Some examples from https://testnet.diva.exchange
- The blocks contain DNS records – stored within «data» commands
- The DNS data entries are isolated by using a simple namespace concept (ns)

Examples (2)

```
[...]
"tx": [
    {
      "ident": "Fr3LwwiP",
      "origin": "9yz3hppnSqz4Oq-6GecT_mTKSNWvYU5oE46yj70SFGw",
      "commands": [
        {
          "seq": 1,
          "command": "data",
          "ns": "IIPDNS:fromcontainer221",
          "d": "fromcontainer221.i2p=nd3kaauoqibfnyujhcht4v3nzahpqztwlyomesfywltuls5bqqi3"
        }
      ],
      "sig": "ATuFW0Stgvz_nF4a3kC4H7XVD67JxlLiqPDIOTcYPR47udiajbyGo3GF70YzvRwtsarAovb69SGL1GlPO2elBw"
    }
],
"height": 142,
"votes":
[...]
```

Summary and Take Outs

- DNS for I2P on a BFT-blockchain is a reasonable solution approach
- The core challenges, as known today:
  - «decision» taking is unstable
  - It's simply «first-come-first-served» (which isn't necessary bad)
- The current implementation is just a prototype
- Partizipation in the project is much welcome - just fork it and get involved please…

Sources

- Container / Docker images, including documentation: https://hub.docker.com/u/divax
- I2P Research (like scalability tests, de-anonymization approaches): https://github.com/diva-exchange/academia
- Democratic Byzantine Fault Tolerance: "Blockchain Scalability and its Foundations in Distributed Systems" by Vincent Gramoli, EPFL, Lausanne, Switzerland, ISBN 978-3-031-12578-2

DIVA.EXCHANGE

Discussion / Links

Web: https://diva.exchange/

Twitter: @DigitalValueX

Mastodon: @social@social.diva.exchange

Telegram Group: https://t.me/diva_exchange_chat_de

Source Code (AGPL3 or better; Apache 2.0) & Research/Academia: https://github.com/diva-exchange

I2P & Docs: https://geti2p.net