

A Generic Data Exchange System for F2F Networks



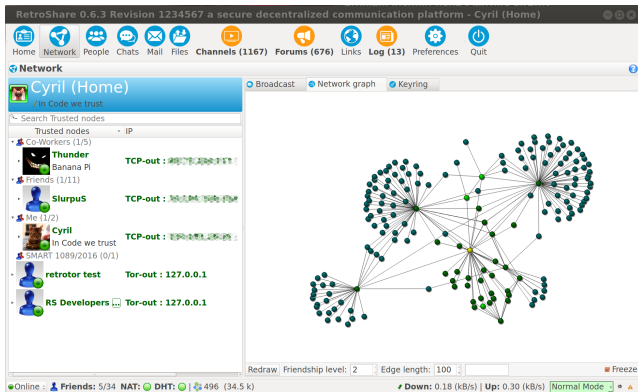
Cyril Soler

Outline

- ▶ Overview of Retroshare
- ▶ The GXS system
- ▶ Decentralize your app!

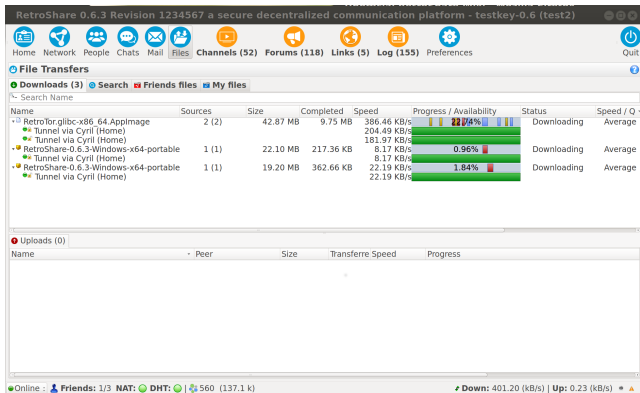
The Retroshare Project

- ▶ Mesh computers using signed TLS over TCP/UDP/Tor/I2P;
- ▶ anonymous end-to-end encrypted FT with swarming;
- ▶ mail, IRC chat, forums, channels;
- ▶ available on Mac OS, Linux, Windows, (+ Android).



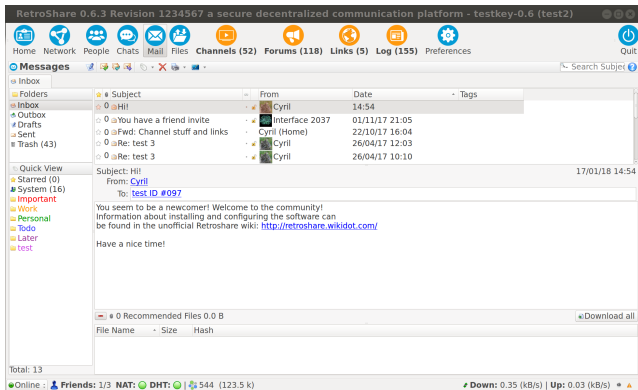
The Retroshare Project

- ▶ Mesh computers using signed TLS over TCP/UDP/Tor/I2P;
- ▶ **anonymous end-to-end encrypted FT with swarming;**
- ▶ mail, IRC chat, forums, channels;
- ▶ available on Mac OS, Linux, Windows.



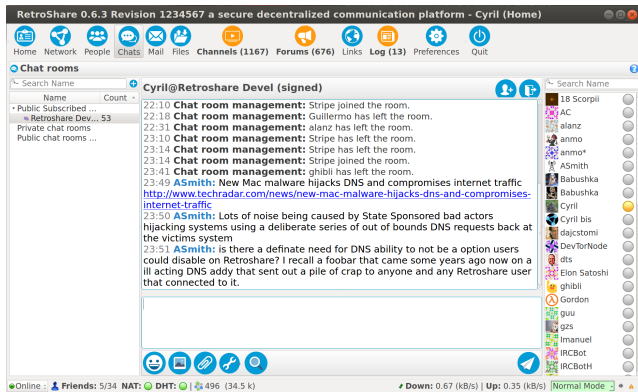
The Retroshare Project

- ▶ Mesh computers using signed TLS over TCP/UDP/Tor/I2P;
- ▶ anonymous end-to-end encrypted FT with swarming;
- ▶ **mail**, IRC chat, forums, channels;
- ▶ available on Mac OS, Linux, Windows.

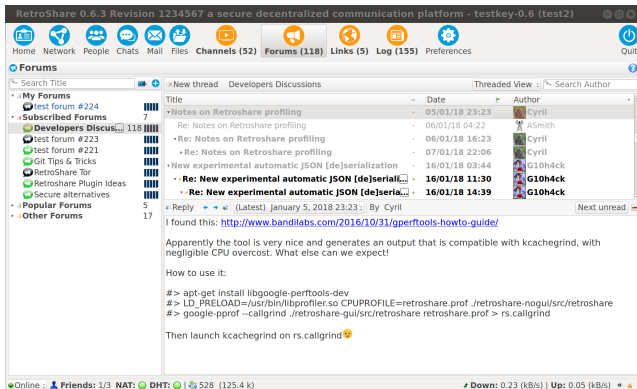


The Retroshare Project

- ▶ Mesh computers using signed TLS over TCP/UDP/Tor/I2P;
- ▶ anonymous end-to-end encrypted FT with swarming;
- ▶ mail, **IRC chat**, forums, channels;
- ▶ available on Mac OS, Linux, Windows.

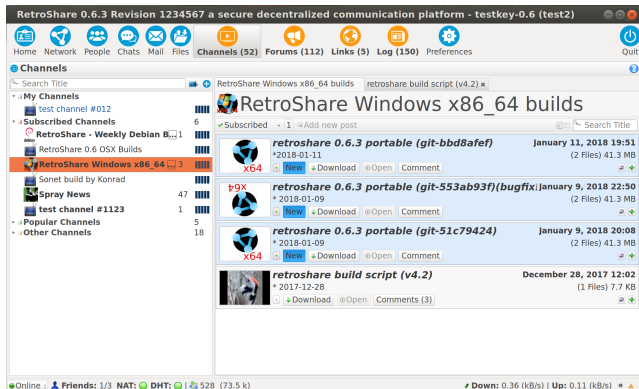


- ▶ Mesh computers using signed TLS over TCP/UDP/Tor/I2P;
- ▶ anonymous end-to-end encrypted FT with swarming;
- ▶ mail, IRC chat, **forums**, channels;
- ▶ available on Mac OS, Linux, Windows.



The Retroshare Project

- ▶ Mesh computers using signed TLS over TCP/UDP/Tor/I2P;
- ▶ anonymous end-to-end encrypted FT with swarming;
- ▶ mail, IRC chat, forums, **channels**;
- ▶ available on Mac OS, Linux, Windows.






The Retroshare Project

History:

- ▶ 10 years old.
- ▶ 5 main contributors (drbob,csoler,G10H4ck,chris,thunder,...)
- ▶ a few thousands daily users (?)

User experience:

-  network bootstrapping is a bit difficult
-  lots of options and possibilities, etc.
-  once you're set, you're pretty much invisible

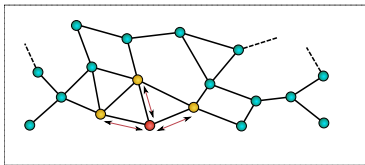
Code:

- ▶ 500,000 lines of C++
- ▶ depends on openssl, libcrypto, OpenPGP-SDK (for now)
- ▶ backend + UI (Qt / Web)
- ▶ channels, forums, email,... : based on a common generic distribution system

Motivation

Friend-to-Friend network:

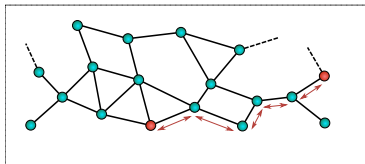
- ▶ mesh of computers connected by authenticated/encrypted links
- ▶ nodes only talk to their trusted neighbors



Motivation

Friend-to-Friend network:

- ▶ mesh of computers connected by authenticated/encrypted links
- ▶ nodes only talk to their trusted neighbors

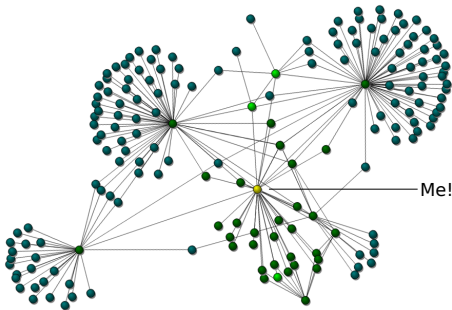


- ▶ publish/exchange data with any node
- ▶ favor interesting content...while preventing flooding, spam, etc.
- ▶ provide authentication/anonymity beyond friends

Motivation

Friend-to-Friend network:

- ▶ mesh of computers connected by authenticated/encrypted links
- ▶ nodes only talk to their trusted neighbors

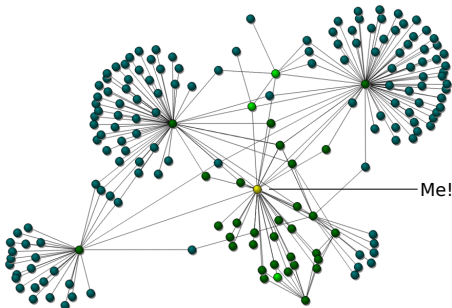


- ▶ publish/exchange data with any node
- ▶ favor interesting content...while preventing flooding, spam, etc.
- ▶ provide authentication/anonymity beyond friends

Motivation

Friend-to-Friend network:

- ▶ mesh of computers connected by authenticated/encrypted links
- ▶ nodes only talk to their trusted neighbors



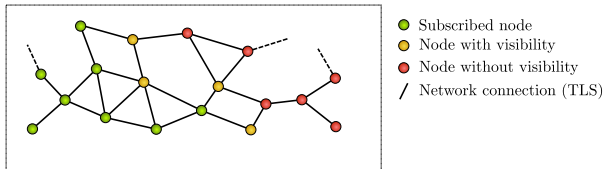
- ▶ publish/exchange data with any node
- ▶ favor interesting content...while preventing flooding, spam, etc.
- ▶ provide authentication/anonymity beyond friends
- ▶ be robust to network changes, disconnections, heterogeneity

Generic eXchange System (a.k.a. GXS)

GXS: Asynchronous distribution, authentication, privacy, security of generic data.

Working principles:

1. subscribers advertise to friends



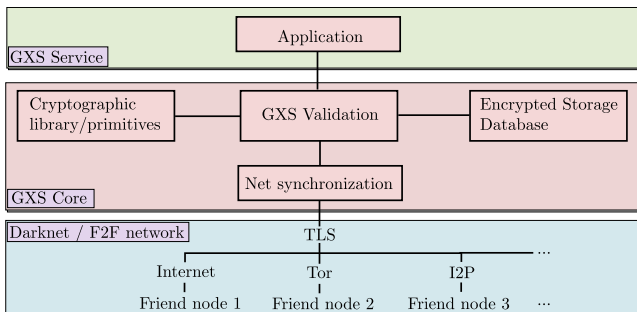
2. nodes team up to ensure data integrity and spam control

Developers implement their own "services/data" on top of it

GXS Core

GXS core automatically provides:

- ▶ local encrypted storage (sqlcipher)
- ▶ network sync.
 - ▶ accounts for access-restriction, storage/sync time periods, etc
 - ▶ multi-chunk transactions
- ▶ validation
 - ▶ data signatures, spam control, cleaning



GXS Core

GXS core automatically provides:

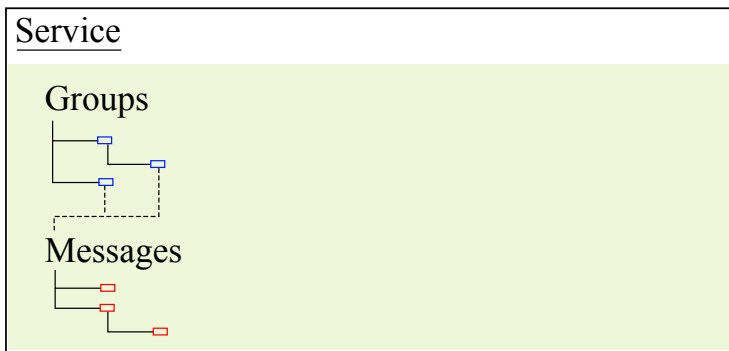
- ▶ local encrypted storage (sqlcipher)
- ▶ network sync.
 - ▶ accounts for access-restriction, storage/sync time periods, etc
 - ▶ multi-chunk transactions
- ▶ validation
 - ▶ data signatures, spam control, cleaning

Specific services implement:

- ▶ private data types (serialization, GUI ↔ GXS types)
- ▶ sync. (auto), subscription (manual) and authentication policies
- ▶ service specific actions

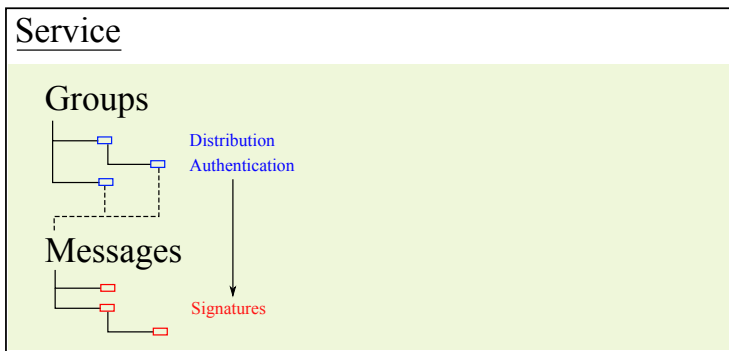
GXS Primitives

Services, Groups, Messages, Identities, Circles



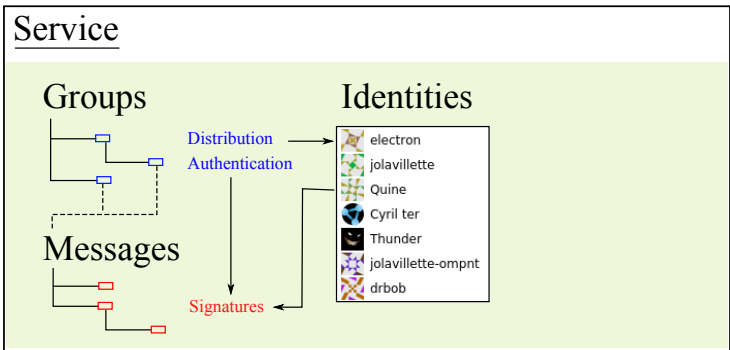
GXS Primitives

Services, Groups, Messages, Identities, Circles



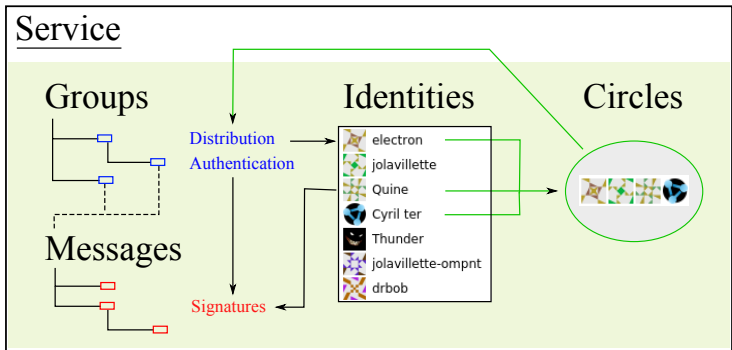
GXS Primitives

Services, Groups, Messages, Identities, Circles



GXS Primitives

Services, Groups, Messages, Identities, Circles



Groups and Messages

- ▶ versionned, hierarchical data
- ▶ meta-data (GXS) + private data (service dependent)

| Group Meta Data | |
|----------------------------|--|
| Field | Type |
| Group Id | 128 bits fingerprint of the public admin key |
| Publish time | 32-bits integer |
| Circle Id | Group Id of parent circle |
| Author Id | Group Id of author identity |
| Description text | Arbitrary string |
| Authentication policy | 32-bits flags |
| Distribution control flags | 8-bits flags |
| Admin key | 2048-bits RSA public key |
| Publish key [optional] | 2048-bits RSA public key |

| Message Meta Data | |
|-------------------|--|
| Field | Type |
| Message Id | 128 bits hash (meta data + private data) |
| Group Id | Id of the parent group |
| Publish time | 32-bits integer |
| Parent Msg Id | Id of parent message |
| Orig Msg Id | Id of previous version of message |
| Author Id | Group Id of author identity |

Pseudo-anonymous identities

- ▶ identities are GXS groups in a "Identity" service
- ▶ sync-ed on request, identities follow groups/messages
- ▶ optionally signed by node key (signature in Group private meta)
- ▶ unsigned identities are anonymous beyond friend nodes


anmo

Identity info

| | | |
|-------------------|--|--|
| Identity name : | anmo |  Send Invite 3 0 |
| Identity ID : | 2d7fdbbc8d6894c131a7588fd05c816fb | |
| Type: | Linked to a known Retroshare node | |
| Owner node ID : | 280BC8B85190BCCE | |
| Owner node name : | anmo | |
| Last used: | 12 minutes ago | |
| Your opinion: | <input checked="" type="radio"/> Positive | |
| Ban-option: | <input type="checkbox"/> Auto-Ban all identities signed by the same node | |
| Friend votes: | 3 positive | |
| Overall: | Positive | |

Usage statistics

12 minutes ago : Membership verification in circle 8f2b21e2bc77f49f840a57a8d1304e30.

Circles

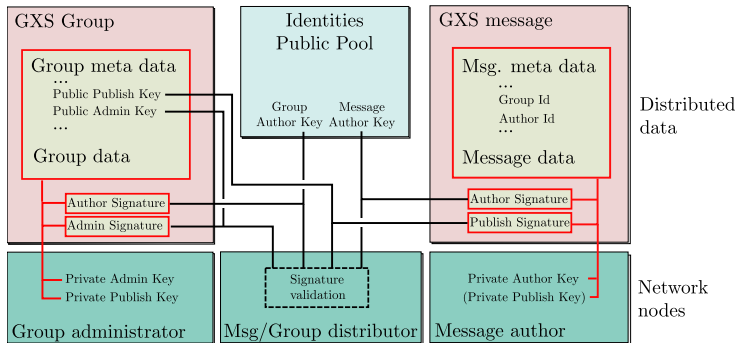
- ▶ Circles are GXS groups in a "Circles" GXS service
- ▶ subscription and sync are automatic
- ▶ membership requires:
 - ▶ invitation: list signed by admin key
 - ▶ membership request: user msg
- ▶ self-restricted circles: only visible to invitee list

| Circle name | Membership |
|--|-----------------------------|
| ◊ Circles I belong to | |
| ◊ test circle #118 | |
| ■ Cyril | Invited |
| ♣ Cyril bis | Invited |
| ■ Interface 2037 | Subscription pending |
| ⚙ test ID #012 | Subscription pending |
| ⚙ test ID #097 | Member |
| ■ Unknown ID :fdd49db341d5ae658251b3b703fb4494 | Invited |
| ◊ test circle #444 | |
| ■ Cyril | Member |
| ⚙ tes ID #009 | |
| ⚙ test ID #097 | |
| ◊ Other circles | |

Circle ID: 66052380f5d1d0c5992e2b55dc402ce6
 Visibility: Public
 Your role: Administrator (Can edit invite list, and request membership).
 Distribution: subscribed (Receive/forward membership requests from others and invite list).
 Your status: Full member (you have access to data limited to this circle)

Data authentication

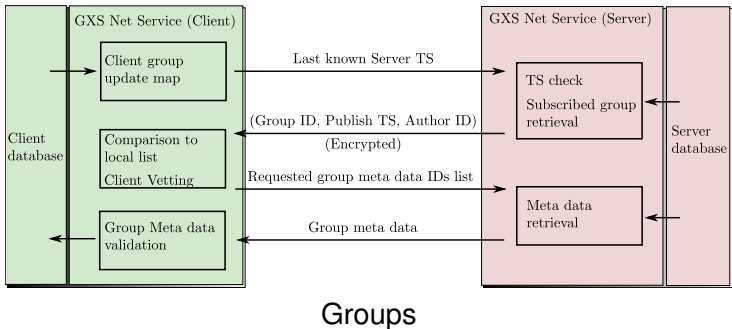
- signature schemes of groups and messages
 - groups: admin, author (depends on service auth. flags)
 - messages: author, publish (depends on Group auth. flags)



Data distribution

► synchronization model

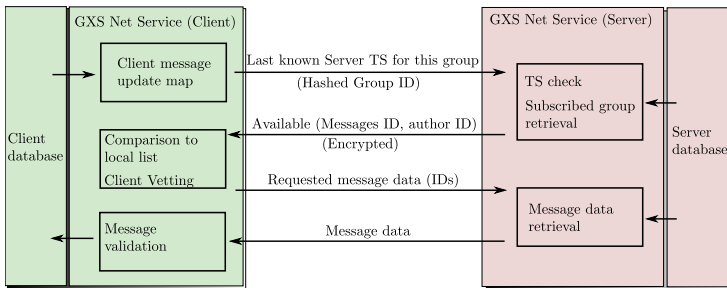
- only compares local times
- circle restriction → data encryption (Anonymized AES+RSA)



Data distribution

► synchronization model

- only compares local times
- circle restriction → data encryption (Anonymized AES+RSA)



Messages

Reputation management

- ▶ Block unwanted content
 - ▶ default settings allow enough visibility
 - ▶ allow newcomers to bootstrap
 - ▶ discourage creation of new identities to spam

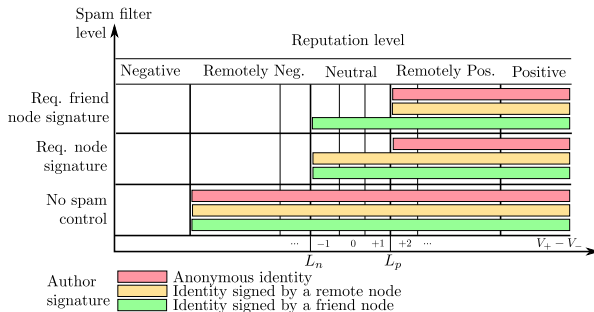
Reputation management

► Block unwanted content

- default settings allow enough visibility
- allow newcomers to bootstrap
- discourage creation of new identities to spam

⇒ always receive data, only forward depending on:

- identity node signature
- opinions sync-ed from friend nodes (local service)
- anti-spam policy for the group

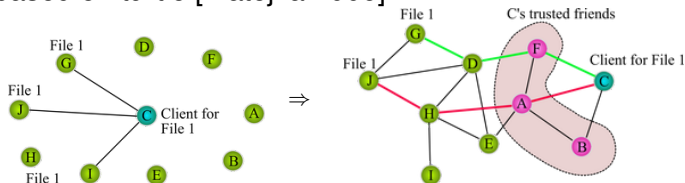


File transfer

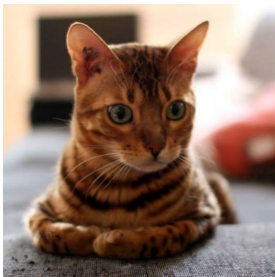
- ▶ Problem: anonymous FT without disclosing data/meta-data to intermediate nodes

File transfer

- ▶ Problem: anonymous FT without disclosing data/meta-data to intermediate nodes
- ▶ tunnels based on turtle [Matejka 2006]



- ▶ no global addressing
- ▶ passive tunnel management
- ▶ multiple tunnels allowed to the same destination
- ▶ anonymity + encryption \Rightarrow needs a pre-shared key
 - ▶ request tunnels using $H(H(f))$
 - ▶ encryption: chacha20+HMAC with $H(H(f)|\text{tunnel_id}|96\text{-bits IV})$



So, what now?

Step-by-step example

Q: So what effort does it take to create e.g. distributed forums?

Step-by-step example

Q: So what effort does it take to create e.g. distributed forums?

A: provide the following (200 lines of code for forums):

Step-by-step example

Q: So what effort does it take to create e.g. distributed forums?

A: provide the following (200 lines of code for forums):

- ▶ service class and authentication policy

```
p3GxsForums::p3GxsForums( RsGeneralDataService *gds, RsNetworkExchangeService *nes, RsGixs* gixs ) :
    RsGenExchange( gds, nes, new RsGxsForumSerialiser(), RS_SERVICE_GXS_TYPE_FORUMS, gixs, forumsAuthenPolicy(),
    RsGxsForums(this), mGenToken(0), mGenActive(false), mGenCount(0)
{
}

uint32_t p3GxsForums::forumsAuthenPolicy()
{
    uint32_t policy = 0;
    uint32_t flag = GXS_SERV::MSG_AUTHEN_ROOT_AUTHOR_SIGN | GXS_SERV::MSG_AUTHEN_CHILD_AUTHOR_SIGN;
    RsGenExchange::setAuthenPolicyFlag(flag, policy, RsGenExchange::PUBLIC_GRP_BITS);

    return policy;
}
```

Step-by-step example

Q: So what effort does it take to create e.g. distributed forums?

A: provide the following (200 lines of code for forums):

- ▶ service class and authentication policy
- ▶ private group/message data and group/message creation code

```
class RsGxsForumMsgItem : public RsGxsMsgItem
{
public:
    RsGxsForumMsgItem(): RsGxsMsgItem(RS_SERVICE_GXS_TYPE_FORUMS, RS_PKT_SUBTYPE_GXSFORUM_MESSAGE_ITEM) {}
    virtual ~RsGxsForumMsgItem() {}
    void clear() { mMsg.clear(); }

    virtual void serial_process(RsGenericSerializer::SerializeJob j, RsGenericSerializer::SerializeContext& ctx)
    {
        RsTypeSerializer::serial_process(j, ctx, TLV_TYPE_STR_MSG, mMsg.mMsg, "mGroup.Description");
    }

    RsGxsForumMsg mMsg;
};

bool p3GxsForums::createMsg(uint32_t &token, RsGxsForumMsg &msg)
{
    RsGxsForumMsgItem* msgItem = new RsGxsForumMsgItem();
    msgItem->mMsg = msg;
    msgItem->meta = msg.mMeta;

    RsGenExchange::publishMsg(token, msgItem);
    return true;
}
```

Step-by-step example

Q: So what effort does it take to create e.g. distributed forums?

A: provide the following (200 lines of code for forums):

- ▶ service class and authentication policy
- ▶ private group/message data and group/message creation code

Comes free with GXS:

- ▶ advertisement of forums to friend nodes
- ▶ distribution of posts to subscribed friends
- ▶ validation of group/message signatures
- ▶ spam control

Application layer (GUI, lots of Qt):

- ▶ creating, visualizing forums/posts
- ▶ editing posts (Meta-data has msg versions)

Step-by-step example

Q: So what effort does it take to create e.g. distributed forums?

A: provide the following (200 lines of code for forums):

- ▶ service class and authentication policy
- ▶ private group/message data and group/message creation code

Subscribed neighbours

Unread posts

Post content
or forum info

Spam control

Post author

The screenshot shows the Retroshare Forums web interface. On the left is a sidebar with a list of forums under the heading 'Forums'. The 'Developers Discussions' forum is selected and highlighted. To the right of the sidebar, a list of posts for 'Developers Discussions' is shown. A red line points from the 'Subscribed neighbours' label to the forum list in the sidebar. Another red line points from the 'Unread posts' label to the 'Unread' column in the post list. A third red line points from the 'Post content or forum info' label to the main content area showing details for the 'Re: Ideas & Todos for 2016' post. A fourth red line points from the 'Spam control' label to the 'Anti-spam' status of the post. A fifth red line points from the 'Post author' label to the author's name 'G10h4ck'.

Forums

- My Forums
 - PGP Signing parties
 - Subscribed Forums
 - RetroShare Support
 - Personal TODO List
 - Developers Discussions
 - Retroshare Plugin Ideas
 - RetroShare Crash reports
 - RetroShare Snapshots
 - Git Tips & Tricks
 - RetroShare Patches
- Popular Forums
 - [F2F-Fr] - Echange de clé pour le...
 - Developer's Discussions
 - RetroShare I2p
- Other Forums

Developers Discussions

| Title | Date | Author |
|------------------------------|----------------|----------------|
| • Re: Ideas & Todos for 2016 | 10/02/16 06:21 | cave |
| • Re: Ideas & Todos for 2016 | 10/02/16 17:49 | sehrat |
| • Re: Ideas & Todos for 2016 | 10/02/16 15:21 | Interface 2037 |
| • Re: Ideas & Todos for 2016 | 03/01/17 16:52 | G10h4ck |
| Re: Ideas & Todos for 2016 | 10/02/16 15:21 | Cyril |

Forum name: Developers Discussions
Subscribers: 9
Posts (at neighbor nodes): 1655
Synchronization: 1 month
Storage: Indefinitely
Distribution: Public
Author: No Signature
Anti-spam: Anonymous/unknown posts forwarded if reputation is positive
Description:
 Discuss here the code, bugs, features, improvements, etc Good place to meet the devs.

Develop fully decentralized apps:

- ▶ Some ideas...
 - ▶ micro-blogging (Twitter)
 - ▶ blogs (pictures, comment threads)
 - ▶ wiki
 - ▶ directory sync
 - ▶ calendar+Tasks
 - ▶ distributed Git
 - ▶ ...
- ▶ Our next target: FB style social network
 - ▶ user's page: GXS group
 - ▶ page posts: GXS subgroups (allows post-based circle visibility)
 - ▶ user's comments: GXS messages in each post group
- ▶ Essentially UI work ;-)
 - ▶ distribution,crypto,...: already done!

Questions?

Sources: <http://github.com/Retroshare/Retroshare>

Developers' blog: <http://retroshareteam.wordpress.com>

Project website: <http://retroshare.net>

Technical report: <https://hal.inria.fr/hal-01617423>

Google Summer of Code 2018

(project ideas here: <https://projects.freifunk.net>)

Thanks to:

