A Dropbox-like Personal Cloud for OpenStack Swift

Pedro García López
Adrián Moreno Martínez
Cristian Cotes González

OpenStack Summit
May 2014 - Atlanta
CloudSpaces project

- Open Service Platform for the Next Generation of Open Personal Clouds:
  [http://cloudspaces.eu](http://cloudspaces.eu)

- Objectives:
  - Interoperability and privacy

- Partners:
  - URV, Eurecom, EPFL, eyeOS (Web Desktop), TISSAT (cloud provider), NEC.
Personal Cloud

We understand it as…

● Storage
● Synchronization
● Sharing
Our motivations

• Not a cloud open-source solution available
  → StackSync and OpenStack

• Users lack control of their data
  → Privacy, Client-side encryption, secure sharing...

• Vendor lock-in
  → Interoperability
Big picture
First stage

- Metadata separated from data
  → StackSync Server
  → OpenStack Swift
Second stage

- MOM (RabbitMQ) ➔ push notifications
- Elastic sync protocol
Current stage

- API as a Swift’s proxy module
- Website
- Android app
Lifecycle and architecture
Lifecycle

StackSync

RabbitMQ

OpenStack Swift

Client 1

Client 2

Client 3
Tasks

- File system watcher
- Chunking, compression, and encryption
Features

- File versioning
- File sharing
- OS integration
  - Overlay icons
  - Notifications
Operations

- `getAccount`
- `getChanges`
- `commit`
- ...
Syncing conflicts

● Two users modify the same file at the same time
● Whichever version is processed first wins
● The loser creates a conflicted copy of the file with his changes
Workspace

- Represents the relation between files and users
- Every user has its own workspace
- Every shared folder has its workspace
- A workspace maps into a Swift container
Encryption settings

Plain  Server-side encryption  Client-side encryption

on your device

on the wire

in the cloud
Swift mappings

StackSync

- installation
- user
- workspace
- admin
- file

OpenStack Swift

- tenant
- user
- container
- tenant admin
- N chunks
Sharing process

1. Send share proposal
2. Notify the addressee
3. Proposal gets accepted
4. Create container and set up ACL
   - X-Container-Read and
   - X-Container-Write

User 1  StackSync Server  User 2

OpenStack Swift
RabbitMQ

- Communication between clients and server
- Push notifications
- Load-balancing
- Multicast messages per workspace
StackSync API

- Metadata DB
- StackSync Server
- RabbitMQ
- OpenStack Swift
- StackSync API
- Desktop client
- Mobile client

Metadata flow

Data flow

Metadata flow (push)
StackSync API

- Located in the proxy’s pipeline
- Used by website & mobile clients
- Communicates with StackSync server to get metadata and authorization
- Activates when the X-StackSync-API header is set
StackSync API Auth

- Module located before the API
- OAuth 1.0a implementation
- Sets the user information in the wsgi environment for the API
Performance
Traffic overhead

![Traffic Overhead Graph]

- Dropbox
- Cloud Drive
- Google Drive
- Box
- StackSync
- OneDrive
Push vs pull
Server elasticity

![Graph showing server elasticity over time](image_url)
Using StackSync
Deployment scenarios

Private
- OpenStack Swift
- StackSync Server

Hybrid
- OpenStack Swift
- StackSync Server

Public
- OpenStack Swift
- StackSync Server

(on premises) (on cloud provider)
Current deployments

- Universitat Rovira i Virgili
- Tissat (Data center)
- Rediris (Spanish university network)
Community

● Available for:
  - Windows
  - Linux
  - Android
  - WWW

● Documentation available at stacksync.org

● Easy to deploy for swifter
Conclusions

● StackSync is a ready-to-use Personal Cloud for OpenStack Swift

● What’s next
  ● Advanced privacy features
  ● Interoperability
  ● iOS and OS X clients
Thanks for your attention!

stacksync.org

github.com/stacksync