

Presenting obsgit

Alberto Planas <aplanas@suse.de>



Agenda

- The Client
- Basic idea
- Comparison
 - OSC + OBS
 - Git + OSC + OBS
- A tour
 - Export
 - Commit
 - Import
- Future

The Client

- The new project requires:
 - Collaboration between different groups to develop a minimal OS ([Carwos](#))
 - A forked version of SLE
 - Frequent cross rebase
 - Tracking of changes
 - Multiple OBS services that are isolated (certifications)
 - True CI/CD with openQA, staging projects and all the heavy stuff
 - Transactional representation of a project (revert)
- **The perfect alibi!**

Basic idea

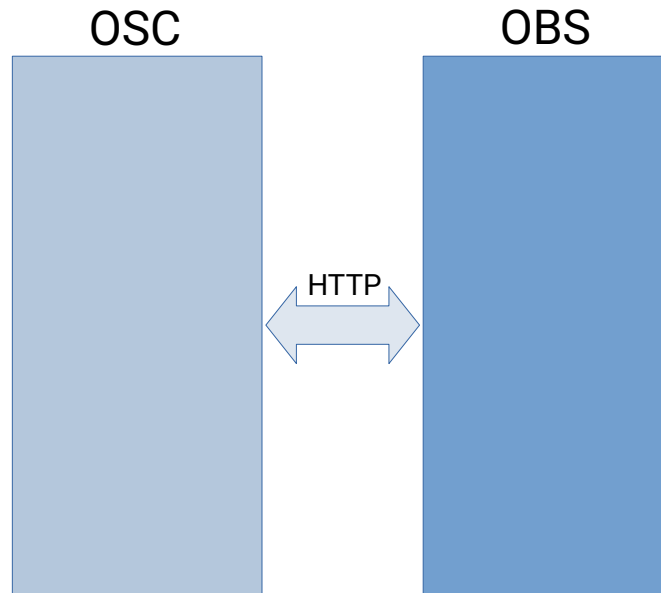
- Separate the build process from the source management
 - Use OBS for building a product definition
 - Use Git for source control
 - Well known and designed for collaboration
 - Common semantic: clone, branch, commit, rebase, {pull, merge} request
 - Consolidated representation of the full project
- ... so, the MVP will be a tool that
 - Export from OBS to Git, including the metadata and the TGZ
 - Import from Git to a different OBS instance

Storage

- Large File Storage (Git LFS)
 - Supported by Gitlab and Github
 - Very fast
 - Require upload in OBS when changes
- OBS
 - No native concept of storage (but we can abstract)
 - Fast
 - API limitations: Cannot copy a single TGZ from one project to another
- [IPFS]
 - Why not!

OSC + OBS

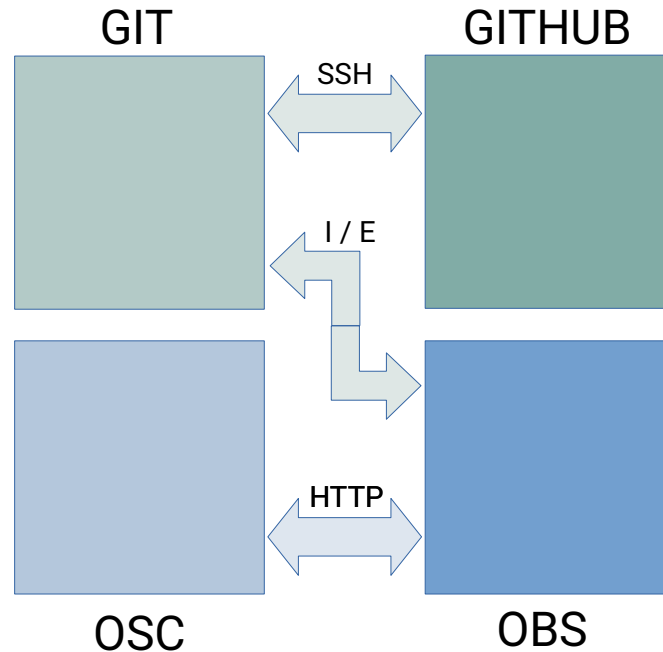
- CLI tool
- SVN based
- Remote commands
- Local build



- Build system
- Product
- Project
- Package
- Architecture

Git + OSC + OBS

- Export
 - Assets
 - Metadata
 - Project
 - Package
- Import
 - Synchronize differences
 - Adjust metadata
- OSC for local builds



- Full source in Git
 - Storage
 - LFS
 - OBS
- Build in OBS
 - Repositories
 - Signature
 - Products

A tour

Export (only one time)

```
$ obsgit export systemsmanagement:yomi yomi
```

```
Initialized the git repository
```

```
Git LFS extension enabled in the repository
```

```
systemsmanagement:yomi/openSUSE-Tumbleweed-Yomi ...
```

```
systemsmanagement:yomi/yomi-formula ...
```

```
systemsmanagement:yomi/yomi-image ...
```

```
systemsmanagement:yomi/salt-bin ...
```

Commit the changes

```
# cd yomi; git status
```

```
On branch master
```

```
No commits yet
```

```
Changes to be committed:
```

```
(use "git rm --cached <file>..." to unstage)
```

```
new file:   .gitattributes
```

```
Untracked files:
```

```
(use "git add <file>..." to include in what will be committed)
```

```
.obs/
```

```
packages/
```

```
# git add .obs packages; git commit -a -m "Initial commit"
```

Edit the files

```
# ls -al packages/yomi-formula/  
.  
..  
.obs  
_service  
_servicedata  
yomi-0.0.1+git.1601999695.6141130.obscpio  
yomi-formula.changes  
yomi-formula.spec  
yomi.obsinfo  
# emacs packages/yomi-formula/.obs/_meta ...  
# git commit -a -m "Important changes"
```

Build the project

```
# obsgit import yomi home:aplanas:yomi-copy
```

```
Git LFS extension enabled in the repository
```

```
home:aplanas:yomi-copy/salt-bin ...
```

```
home:aplanas:yomi-copy/openSUSE-Tumbleweed-Yomi ...
```

```
home:aplanas:yomi-copy/yomi-formula ...
```

```
home:aplanas:yomi-copy/yomi-image ...
```

Future

- Fully separate the build system from the source management
- Better storage in OBS
- git-OBS (rewrite as a Git plugin)
 - `git obs export`
 - `git obs build`
 - `git obs sr`
- Export the history of a package?
- Avoid monorepo?
 - A `git clone openSUSE/Factory` will be too much
 - Re-use the concept of devel projects?
 - Git submodule or Git subtree
- 1:1 feature with current model
 - How to map a `_link`?

2020



Thank You



All text and image content in this document is licensed under the Creative Commons Attribution-Share Alike 4.0 License (unless otherwise specified). "LibreOffice" and "The Document Foundation" are registered trademarks. Their respective logos and icons are subject to international copyright laws. The use of these thereof is subject to trademark policy.