A Solution to the Backup Inconvenience

Fast, Secure and Efficient Backups with "restic“

Alexander Neumann
Florian Daniel

FrOsCon 2015 – 23 August 2015

Agenda

- Project background
- restic design principles
- Features & key components
- Demonstration & technical background
- Project & contribution
- Q&A

Project Background

Why another backup solution?

- Shared backup location – server in „partly“ secure location
- Tested a variety of backup solutions – none satisfactory
- Use case: 16GiB, 140k Files, 36k Directories
- Most solutions slow, particularly over WAN connections
- Some no crypto or no deduplication
- Usability cumbersome

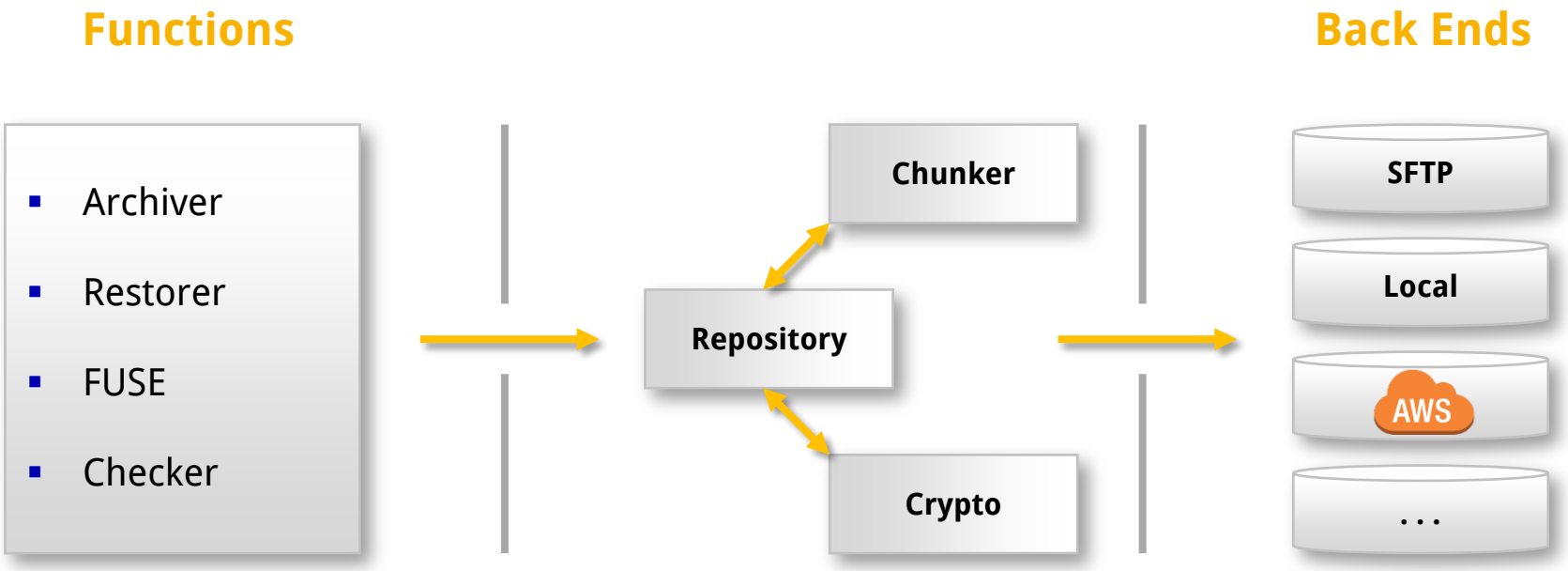
restic Design Principles

Easy	▪ Easy frequent backups ▪ Automation
Fast	▪ High throughput ▪ Physical limitations only
Verifiable	▪ Validation on backup location without decrypting ▪ Integrated verification function
Secure	▪ Full encryption (data, metadata) ▪ Backup to non-trustworthy locations ▪ Authenticate backup during restore (detect tempering)
Efficient	▪ Deduplication ▪ No regular full backup required as a basis for increments
Free	▪ Free Software ▪ Transparent development process

Features

- Supported OS (Linux, OS X, *BSD, Windows)
- Deduplication
- Stable yet flexible repository format
- Very fast (although still improving)
- Multiple machine backups to one repository
- Pluggable back ends (local, SFTP, AWS S3, ...)
- Browse backups (FUSE support, embedded webserver)

restic Components

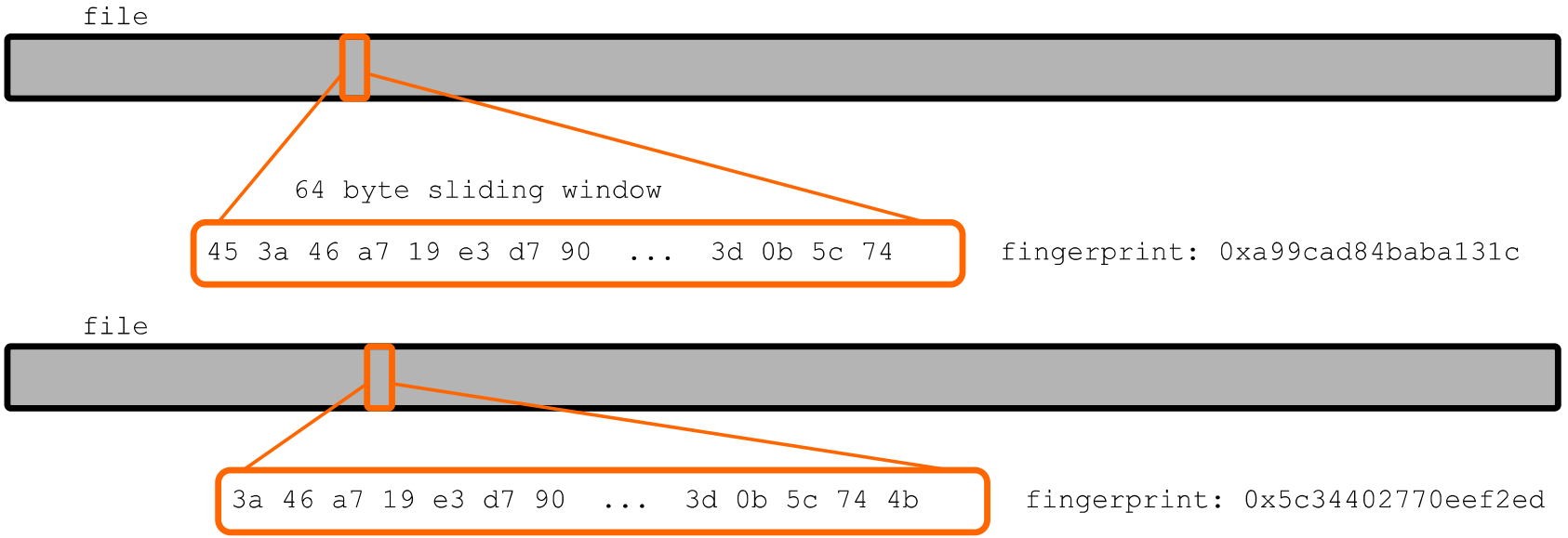


Demonstration

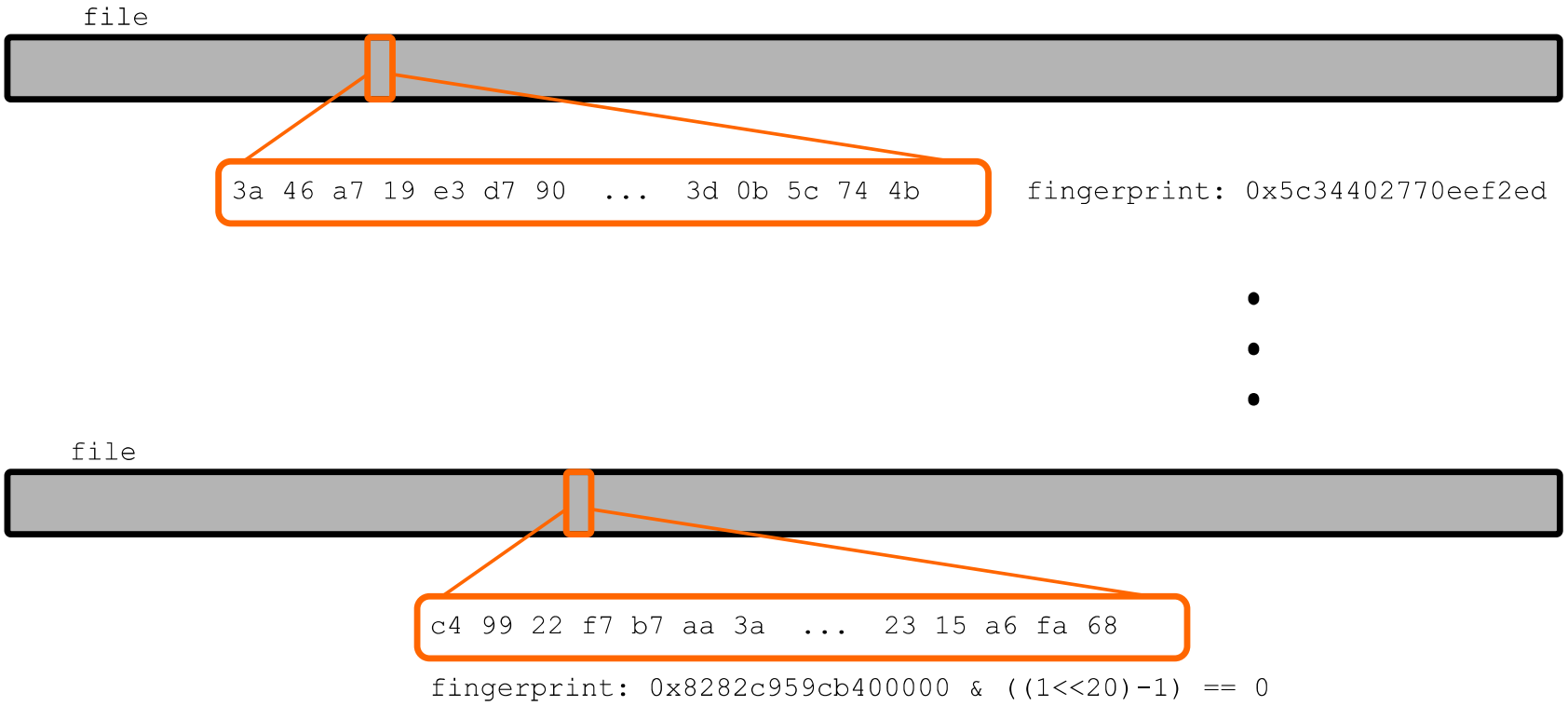
restic Implementation

- Written in Go (>=1.3)
- Crypto: AES-GCM / Poly1305-AES
- Content Addressable Storage (like in Git/Camlistore)
- Metadata: JSON
- KDF: scrypt
- Content Defined Chunking (Rabin Fingerprinting)

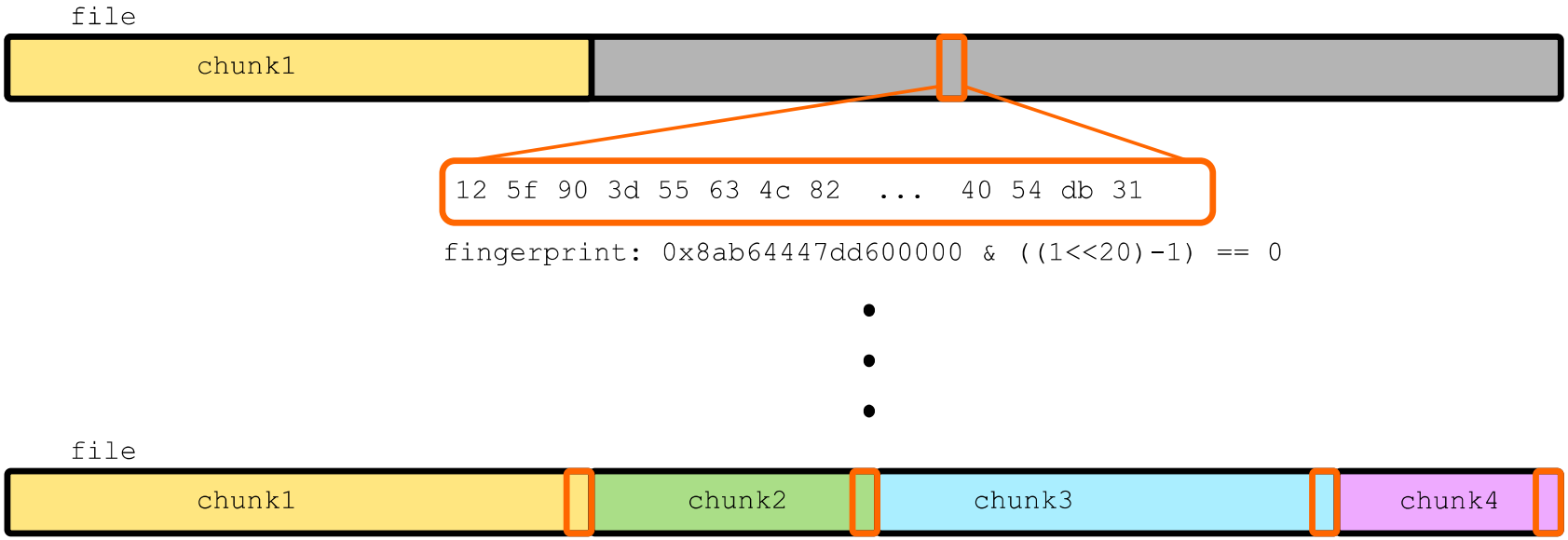
Content Defined Chunking #1



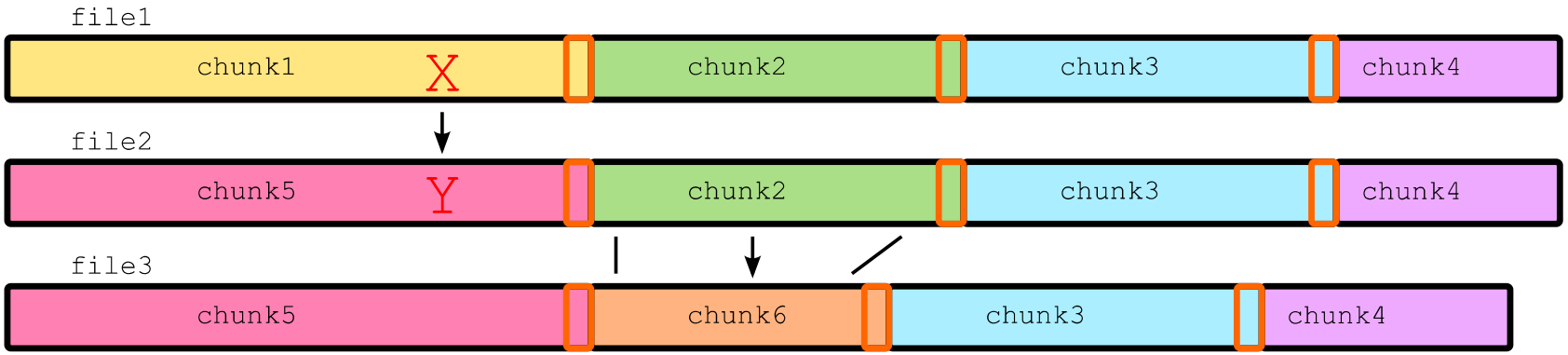
Content Defined Chunking #2



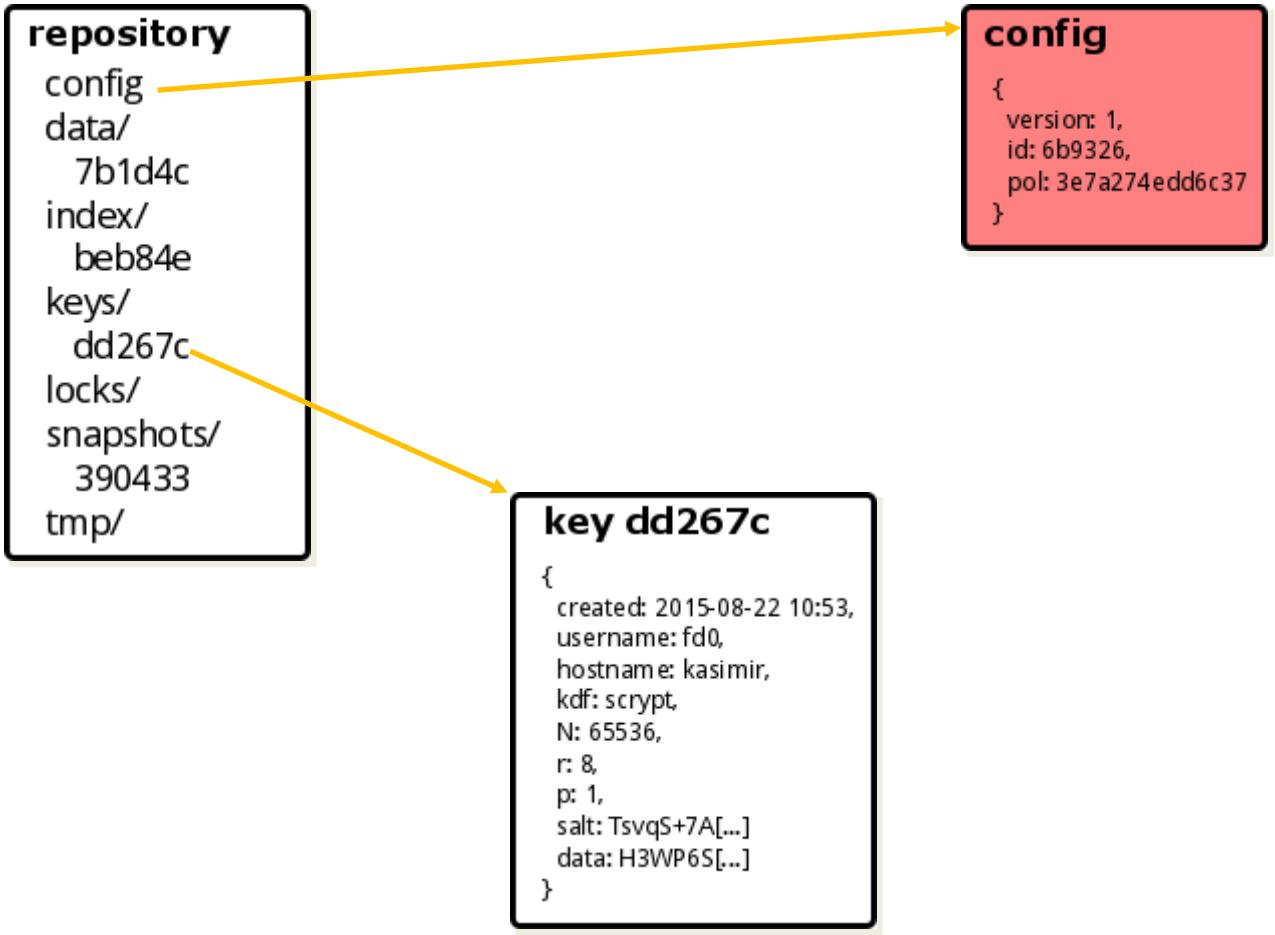
Content Defined Chunking #3



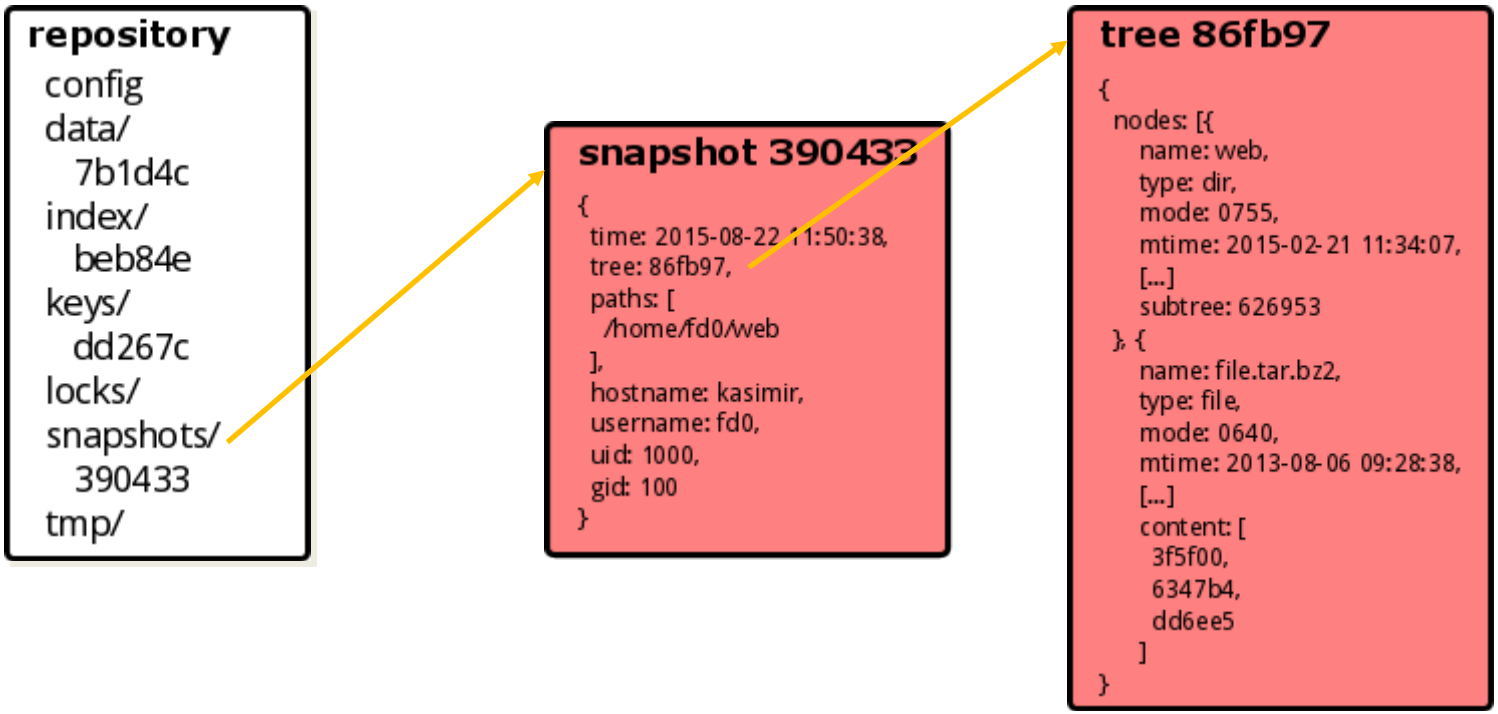
Content Defined Chunking #4



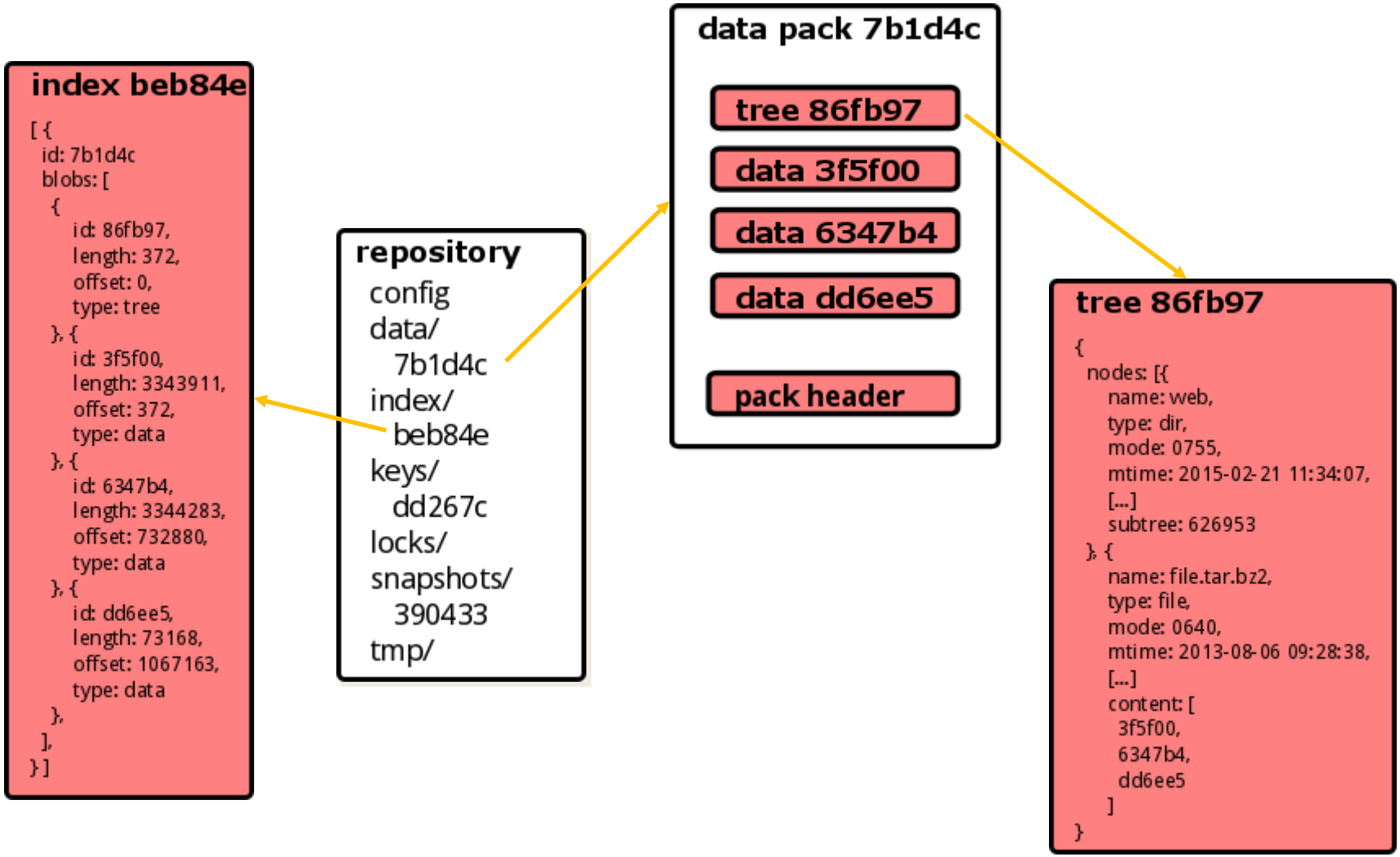
restic Repository Structure



restic Repository Structure



restic Repository Structure



Demonstration

Current Status and Outlook

- Current release: 0.1.0
- Repository format stable

- More back ends (Google CS Nearline, AWS Glacier)
- User interface improvements
- Compression
- Resume interrupted backup
- Performance: add more concurrency
- Backup: asymmetrical crypto
- Backup from stdin (mysqldump, etc.)

Getting Started

```
$ git clone https://github.com/restic/restic
$ cd restic
$ go run build.go

$ ./restic -r /srv/backup init
$ ./restic -r /srv/backup backup ~/work
```

Project & Contribution

- Free Software (BSD 2-Clause License)
- Repository: https://github.com/restic/restic
- Priority management: https://waffle.io/restic/restic
- 3 Core contributors / 12 contributors

How you can help:

- Install & use restic
- Report bugs
- Request features

Questions?

restic.github.io

Thank You