



@schlomo



@schlomo



@schlomoschapiro



@schlomo@floss.social



# Cloud & Offline Secrets Management

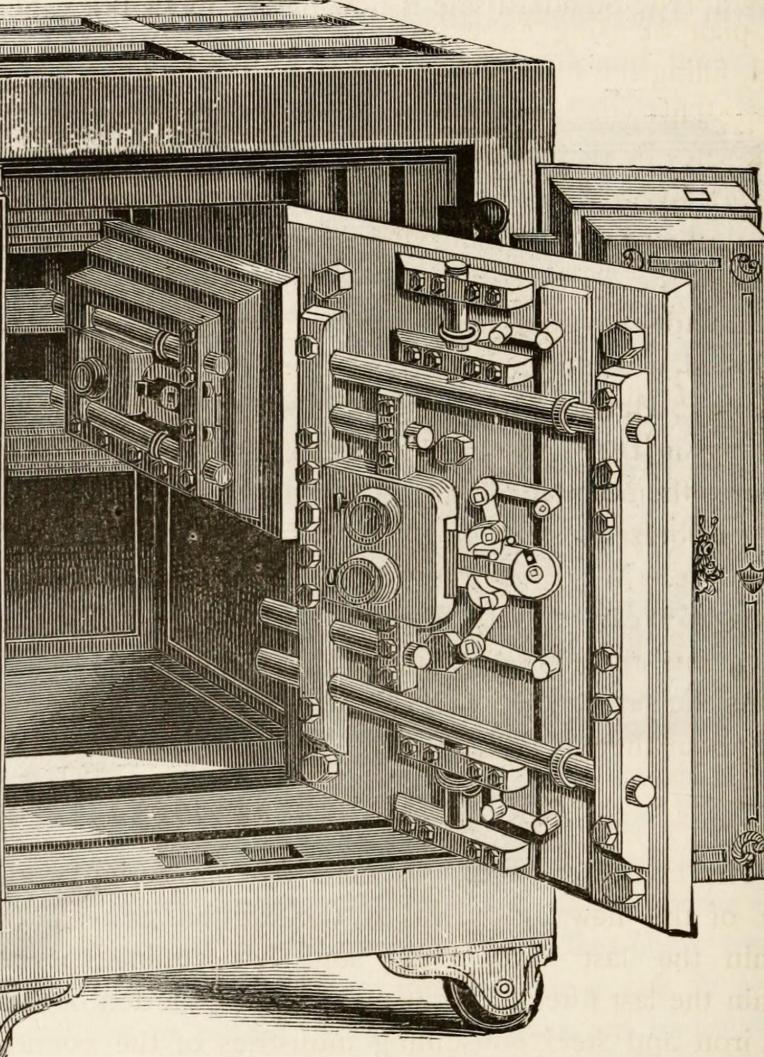
## Managing operational secrets with SOPS

Schlomo Schapiro, Associate Partner / Principal Engineer, Tektit Consulting  
19.06.2024, DevOpsCon 2024, Berlin

DevOpsCon by devmio



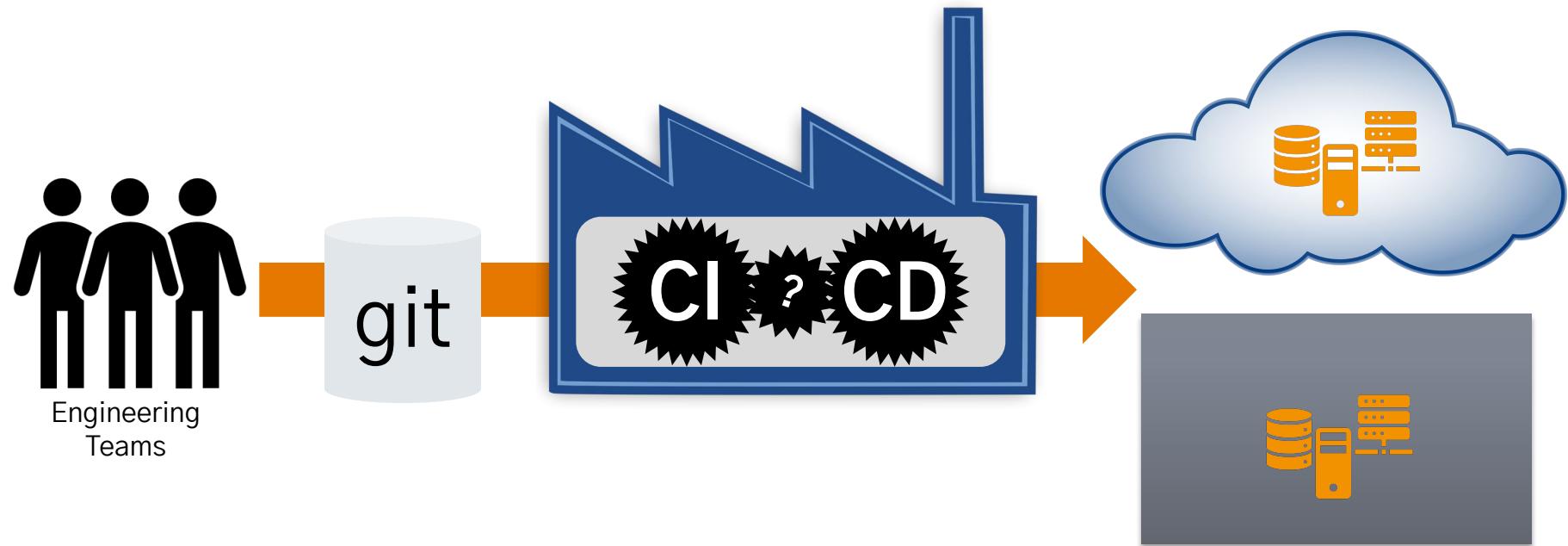
This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License. Original author: Schlomo Schapiro (Forte)



# Agenda

1. Context: DevOps
2. Why Secrets?
3. Functional Requirements
4. Non-Functional Requirements
5. What Could Possibly Go Wrong?
6. SOPS – Secrets OPerationS
7. Backup & Disaster Recovery

# Happy DevOps Campers



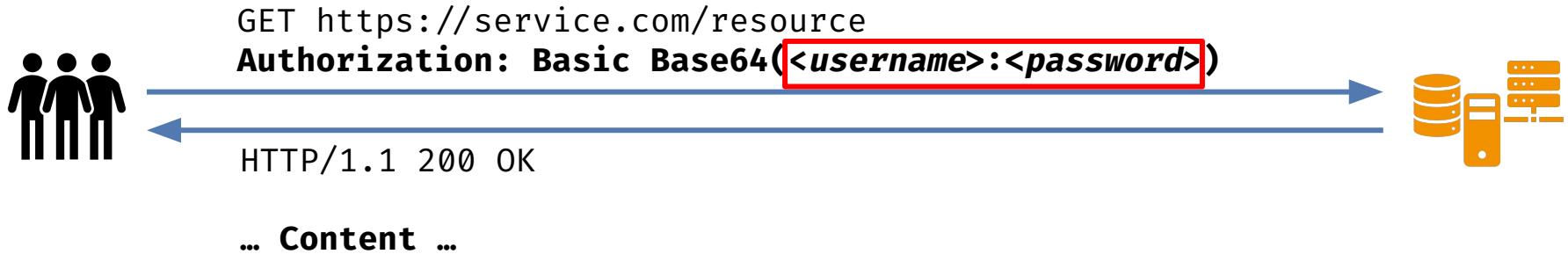
# DevOps is

- … if every person uses the same tool for the same job
- … codified knowledge – everybody contributes his part to common automation
- … if all people have the same privileges in their tooling
- … if human error is equally possible for Dev and Ops
- … replacing people interfaces by automated decisions and processes

[bit.ly/5devops](https://bit.ly/5devops)

… a result

# Why Secrets?



Read more in my blog at [schlomo.schapiro.org](http://schlomo.schapiro.org)

1. Lifting the Curse of Static Credentials

[schlomo.schapiro.org/2016/05/lifting-curse-of-static-credentials.html](http://schlomo.schapiro.org/2016/05/lifting-curse-of-static-credentials.html)

2. Eliminating the Password of Shared Accounts

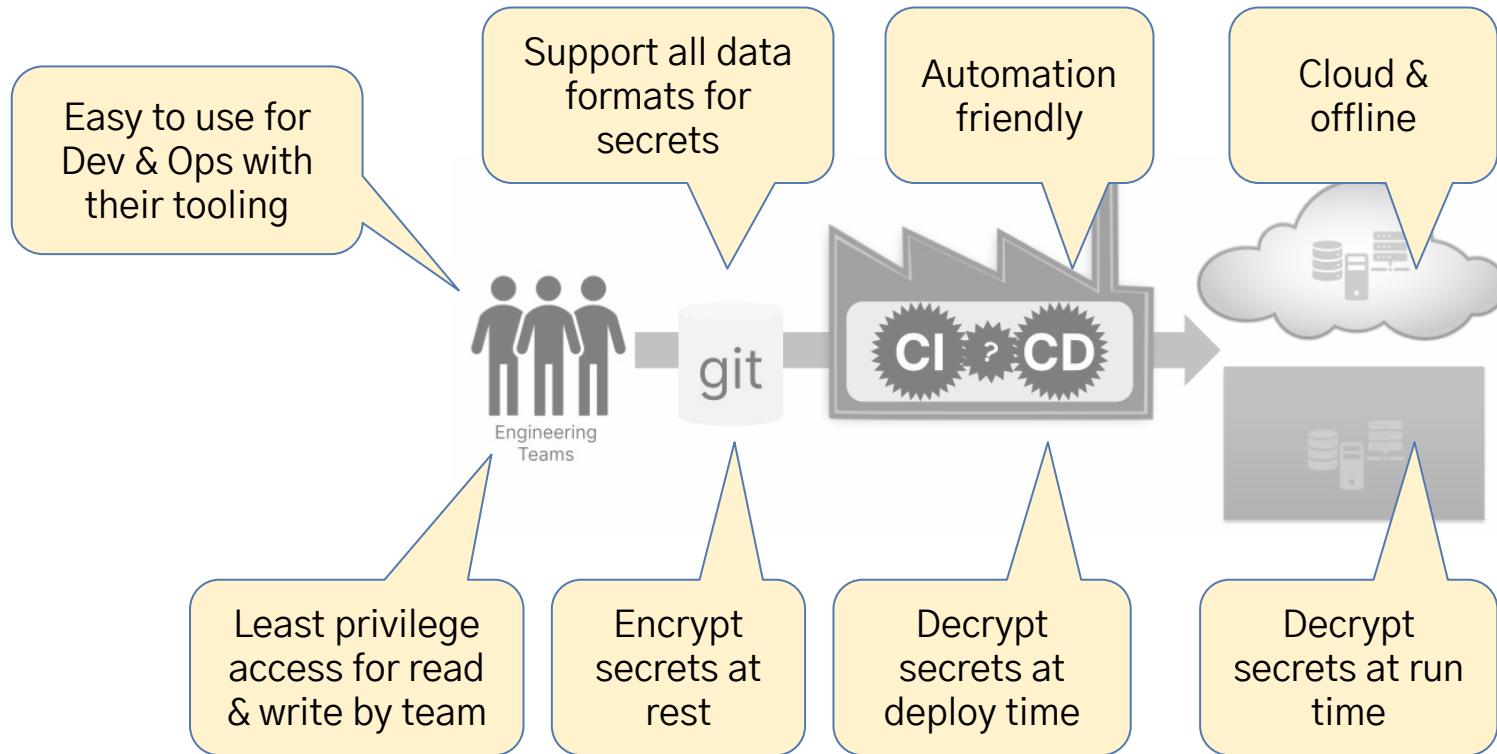
[schlomo.schapiro.org/2017/06/eliminating-password-of-shared-accounts.html](http://schlomo.schapiro.org/2017/06/eliminating-password-of-shared-accounts.html)

3. A Login Security Architecture Without Passwords

[schlomo.schapiro.org/2022/02/login-security-architecture-without-passwords.html](http://schlomo.schapiro.org/2022/02/login-security-architecture-without-passwords.html)



# Functional Requirements for Secrets Management



# Non-Functional Requirements for Secrets Management

- Strong identity verification of users and deployment / runtime software
- Stolen or lost laptop doesn't pose a risk
- Immediate off-boarding of users if needed, cannot retain access to copied secret stores
- Reduce the exposure of secrets by segmenting secrets access per team, department or criticality / blast radius, as much as reasonably possible
- Prevent tampering with secrets by separating between decryption permissions used for software deployment and encryption permissions used by engineers
- Change management and traceability of changes at least as good as for software and configuration changes
- Secrets management should have no or only limited impact on operational ability to effect changes in production, e.g. perform a deployment or change configuration
- Retain access to secrets under all circumstances, even if we lose access to one or all Cloud accounts or services
- ...

# What Could Possibly Go Wrong?



All my Data is  
in the Cloud!



10.03.2021: [OVHcloud data centre destroyed in inferno](#)

Where is my  
Cloud Data?



# Google refuses to reinstate man's account after he took medical images of son's groin

Experts say case highlights dangers of automated detection of child sexual abuse images



Tech companies like Google have access to a vast trove of data – but no context for it, says an ACLU technologist. Photograph: Avishek Das/Sopa Images/Rex/Shutterstock

22.08.2022: [Google account is lost for good \(The Guardian\)](#)



# The Problem: Users are Responsible for Content

Accidentally or maliciously deleting data?

Granting access to malicious apps?

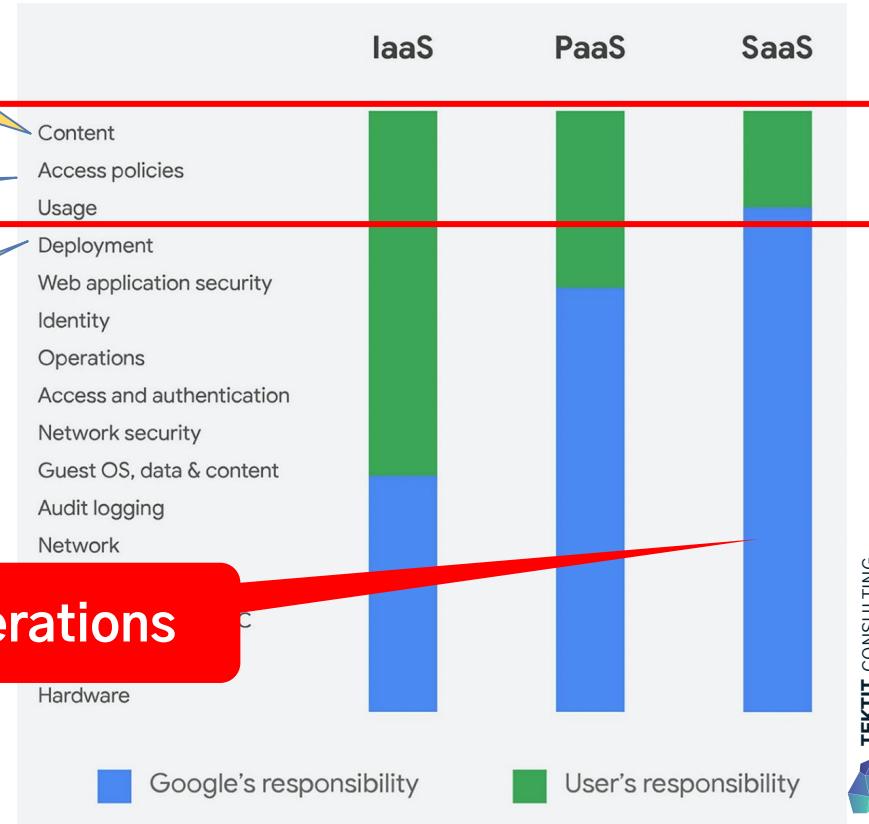
Deleting entire user account or Domain?

Vendor only guarantees technical operations

Example Source:

[Google Workspace data protection implementation guide](#)

Released 12/2020





## Mission Impossible: Complete Google Workspace Disaster Recovery

- Commonly used SaaS for collaboration, communication & office productivity
- **Data ownership ≠ data possession**
- NO complete backup possible!
- Only **partial** backups possible!
- Everybody accepts the risk!

See [Mission Impossible:  
Complete Disaster Recovery for  
Google Workspace](#)



# Secrets OPerationS



“SOPS (Secrets OPerationS) is an **editor** in the form of a **command-line** tool and **SDK** designed to help **manage encrypted files** in a variety of structured (YAML, JSON, ENV, INI) and BINARY formats using one of the supported **Key Management Systems (KMS), PGP, or age.**”

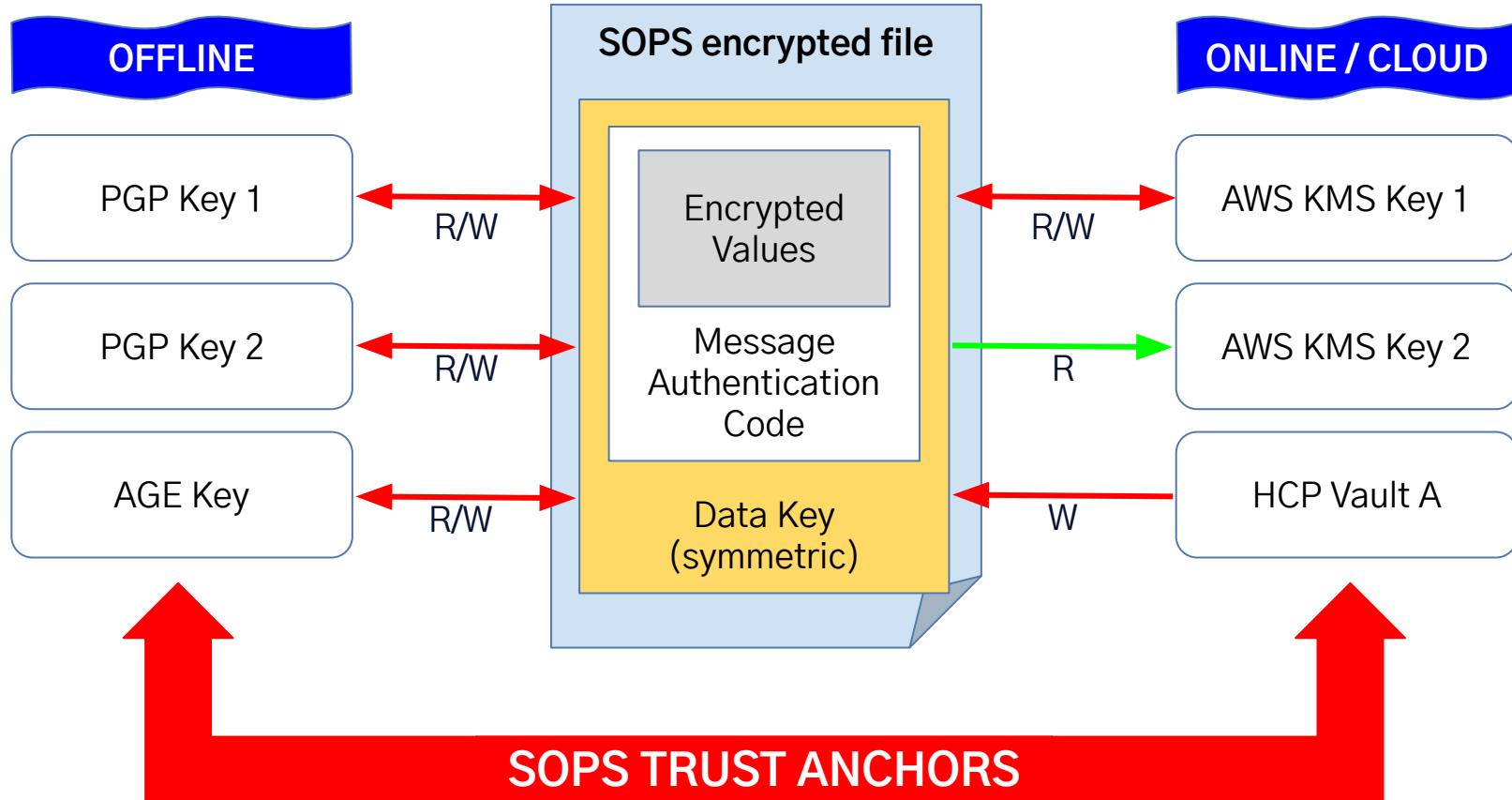
*Source: getsops.io*

Method	Encrypt	Decrypt
<b>Offline:</b> PGP/GPG, age	Public Key	Private Key
<b>Cloud:</b> KMS (GCP KMS, AWS KMS, Azure Key Vault, Hashicorp Vault)	Encrypt Permission	Decrypt Permission

**age** - A simple, modern and secure encryption tool (and Go library) with small explicit keys, no config options, and UNIX-style composability.



# Secrets OperationsS Architecture



1300-467-7733 (ext.) • [www.psu.edu](http://www.psu.edu)

[View details](#)

<http://www1.umn.edu/tlrc/tlrcweb/42000.htm>

Рис. 4303714 Запасные части для станка с ЧПУ 1720 1048180471

CH-2005-000-1440003204,4200

—WESTIN HOTELS  
INTERCONTINENTAL HOTELS

# Secrets OperationsS

```
info: Welcome to SOPS! Edit this file as you please!
example_key: example_value
# Example comment
example_array:
  - example_value1
  - example_value2
example_number: 1234.56789
example_booleans:
  - true
  - false
```

# Secrets Operations - encrypted file

```

info: ENC[AES256_GCM,data:HYGEJN0q3C6c4Id6d9CE40Va15mX/8uE+M2Dr050Nd2hDTpUKw5oNEJ
example_key: ENC[AES256_GCM,data:h4ZPZQVP3V3hVxt6Mw==,iv:x8mYCxxpzWBbN5sf0fr2V5IB
#ENC[AES256_GCM,data:6gX78q+XkdVTGYd1CHxXCw==,iv:ce5lH6voUQnea70Ksu1DWSAgKTgZ7m0h
example_array:
  - ENC[AES256_GCM,data:HC5zVU6LaVzehk77Hos=,iv:6C/pusncdpKGZFTX569+5lVRkoJHNhsI
  - ENC[AES256_GCM,data:r6DuIBIn+mbi70M2f2E=,iv:fNTW4iWd4rt98zqnw81D2fNBnARt+C7c
example_number: ENC[AES256_GCM,data:3xKjcH9GJ06Zdw==,iv:ISJTxCs+ITs8+XUch45a/w5Mo
example_booleans:
  - ENC[AES256_GCM,data:mpAj/A==,iv:S+3cL9klQ/3D4Waa1kXz3RBF68nhZDV4CHuPF0Zc84I=
  - ENC[AES256_GCM,data:NG0xinc=,iv:Tj9bSL5d1Hlx5yAZ07jpyNL3keVYAvUJi9VNDNcD0B4=
sops:
  kms: []
  gcp_kms: []
  azure_kv: []
  hc_vault: []
age:
  - recipient: age12pewudxq53khcgm49flqq7t6l5na8jscsnn4lqyxla4nzzm4l92qsk7qq
    enc: |
      ——BEGIN AGE ENCRYPTED FILE——
      Vwd11WVvUY31FchRpnb24uh2JnL2YxGio+TEgyNTUxOSRUNJNRgE+NT01nPvZxOzPK

```

# Secrets Operations - encrypted file explained

## example\_array:

- ENC[AES256\_GCM,data:HC5zVU6LaVzehk77Hos=,iv:6C/pusncdpKGZFTX569+5l...]
- ENC[AES256\_GCM,data:r6DuIBIn+mbi70M2f2E=,iv:SNTWLiWk+100...]

Encrypted Values

## sops:

kms: []

Plaintext Keys

gcp\_kms: []

azure\_kv: []

hc\_vault: []

Trust Anchor ID

## age:

- recipient: age12pewudxq53khcgm49flqq7t6l5na8jscsnn4lqyxla4nzzm...

enc: |

Encrypted Data Key

----- BEGIN AGE ENCRYPTED FILE -----

YWdlLWVuY3J5cHRpb24ub3JnL3YxCi0+IFgyNTUxOSBUNlNPaEtNT01nRVZx...

...

----- END AGE ENCRYPTED FILE -----

Tamper Proofing

lastmodified: "2023-11-14T13:06:19Z"

mac: ENC[AES256\_GCM,data:GcnG90R58Se0f06kukMUiBfB8MJ+SnB2RgXJKqBvMK1...

# More about SOPS → [getsops.io](https://getsops.io)

## Excellent tooling support:

- VS Code plugin, IntelliJ plugin,
- Terraform provider, wrapper, … and Ansible integration
- Lots of Kubernetes tooling supports SOPS
- Configure SOPS standard keys and behaviour via .sops.yaml file
- ...

SOPS

## Advanced security features:

- Key rotation via sops -r
- Require multiple master keys (key groups) via --shamir-secret-sharing-threshold
- Unencrypted values via --unencrypted-suffix or --unencrypted-regex
- diff support for git diff ...
- Encrypt binary files
- Upload encrypted files to S3, GCS ...
- Audit trail



# SOPS Usage

**Configure:** Create `.sops.yaml` with default settings and **trust anchors**:

```
creation_rules:
  - path_regex: secret
    age: age12pewudxq53khcgm49flqq7t6l5na8jscsnn4lqyxla4nzzm4l92qsk7qq4
```

**Encrypt:**

```
sops secrets.env
```

**Decrypt:**

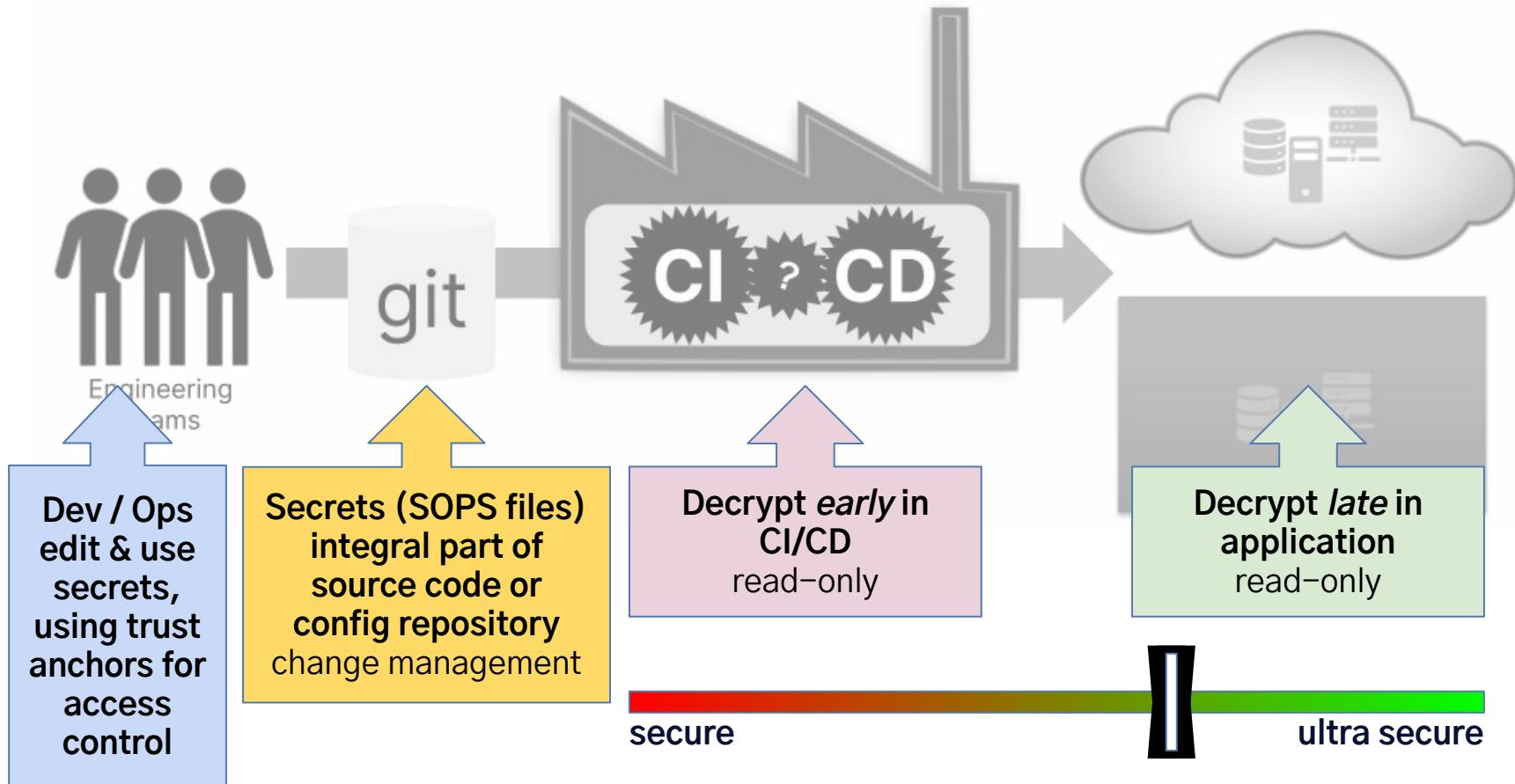
```
export SOPS_AGE_KEY=AGE-SECRET-KEY-165DJSTUXKL8WEUEJJ9H3M25YKQUQ3RDGTQJJ9YU72PK3F6NZ26NQRD6NRT
```

```
sops -d secrets.env
```

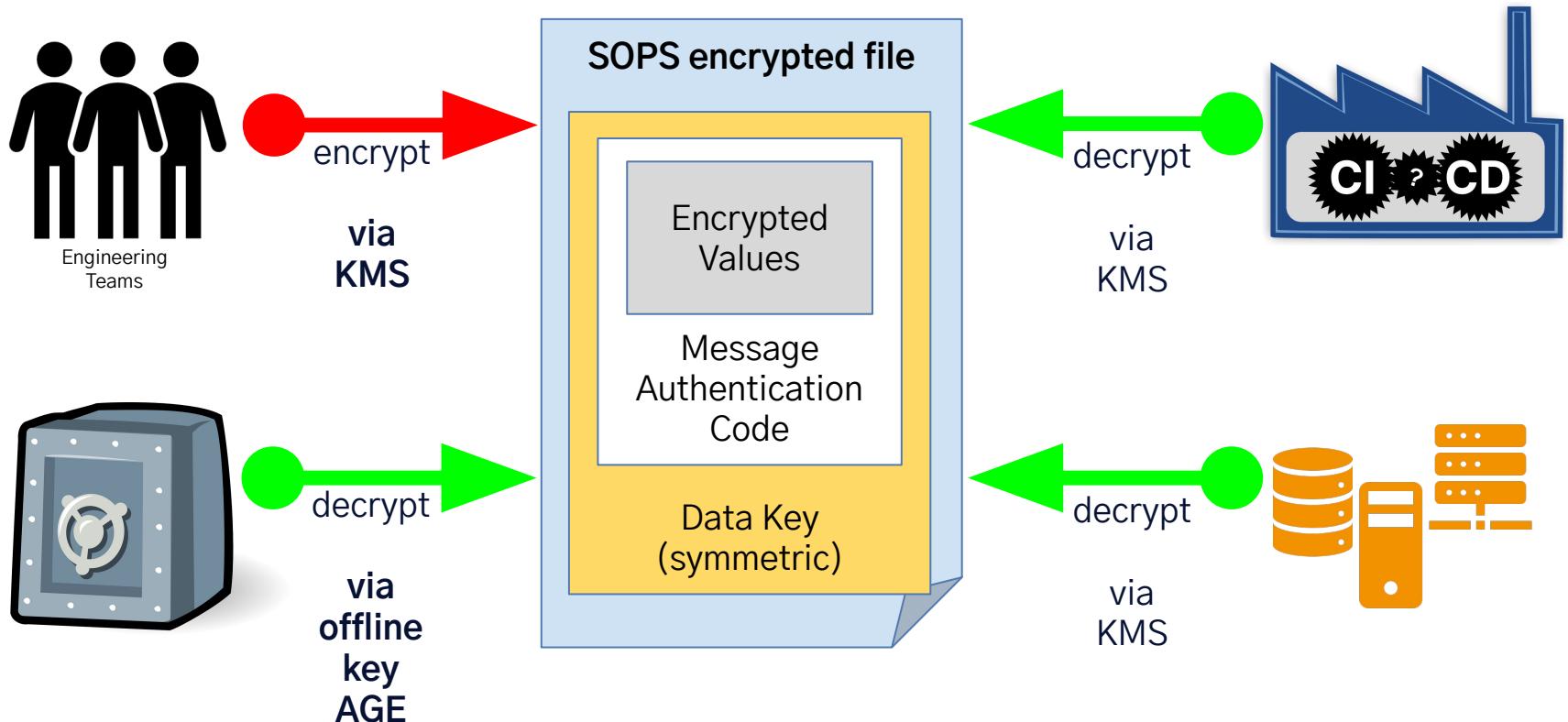
```
sops exec-env secrets.env ./run.sh
```

```
sops exec-file secrets.env './run.sh --secrets {}'
```

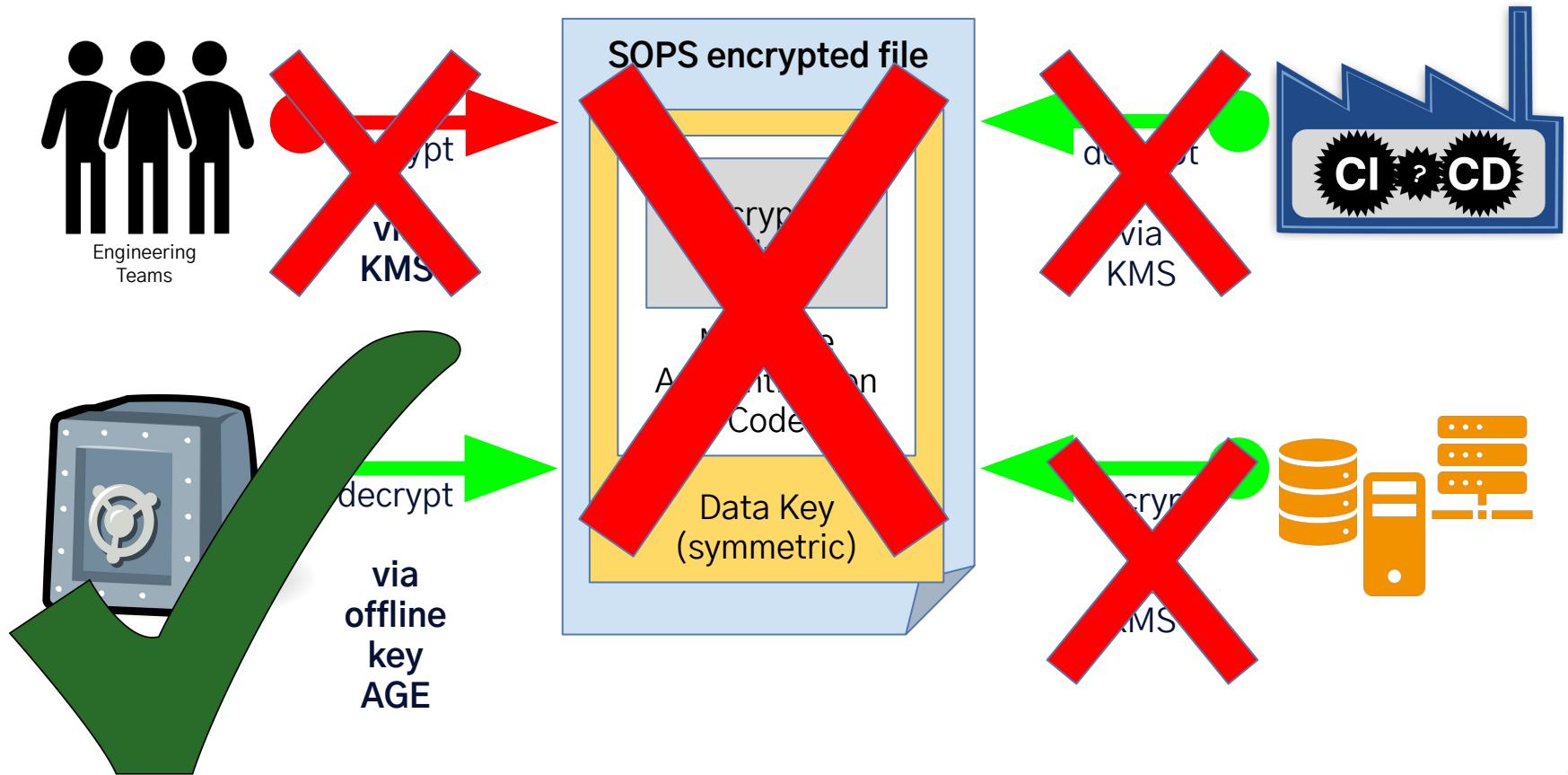
# SOPS in the Software Delivery Life Cycle



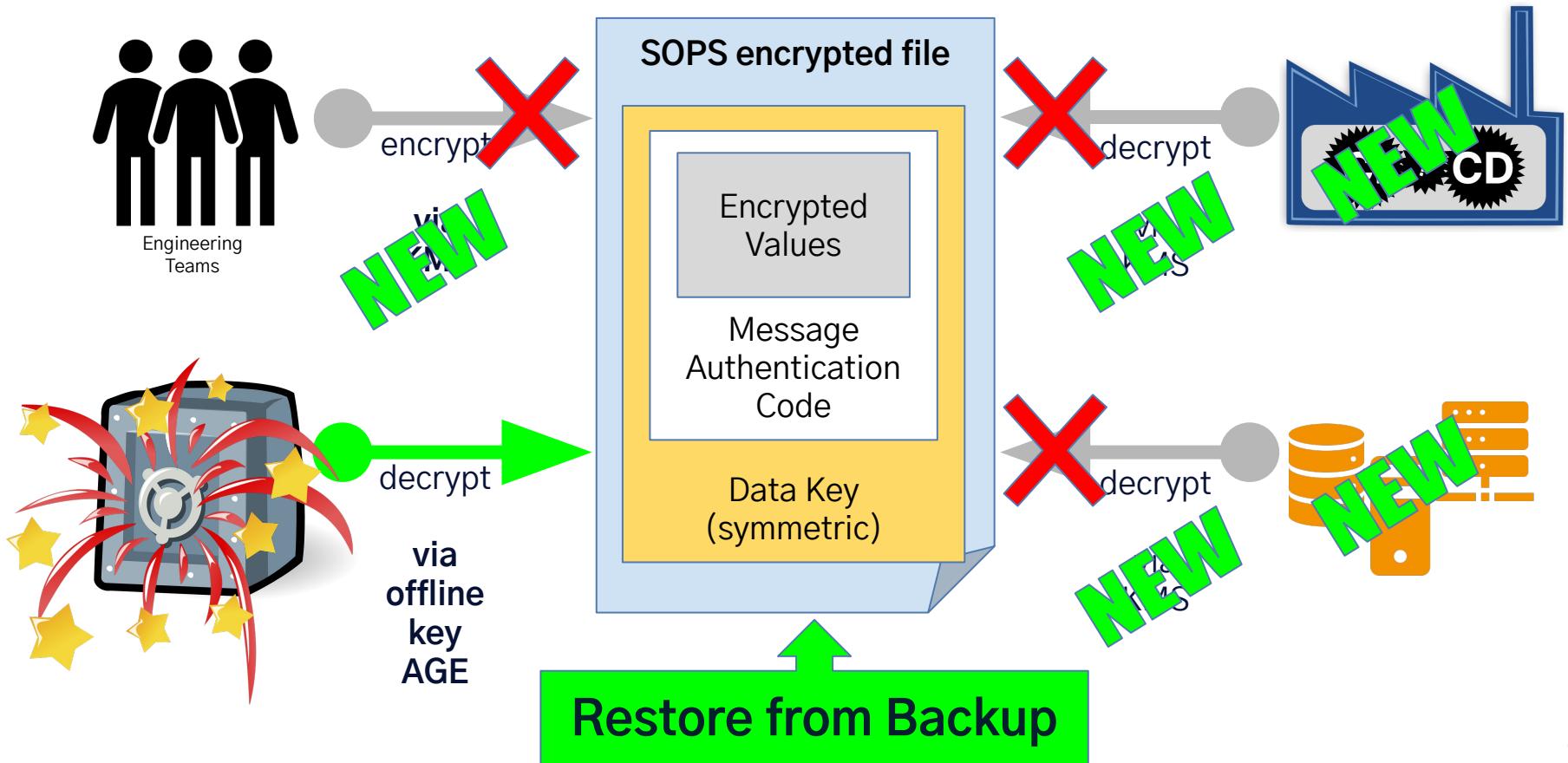
# SOPS Trust Anchors → “Secrets Management”



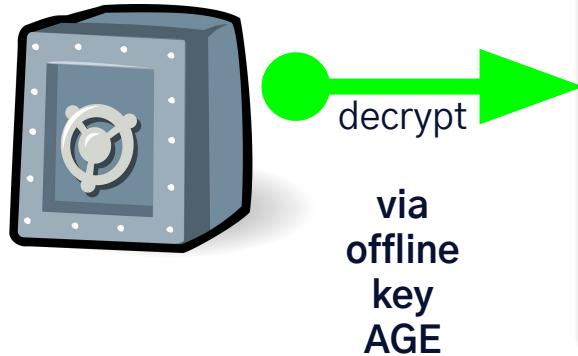
# Disaster — All Cloud Data or Services are Gone!



# Recovery — All Cloud Data and Services are New!



# Recovery — Restore Access for New SOPS Trust Anchors



1. Update .sops.yaml with **new** trust anchors
2. sops updatekeys -y FILE
- 3.

```
› sops updatekeys demo.env -y
```

```
2023/11/15 17:01:09 Syncing keys for file /Users/schlomoschapiro/Downloads/demo.env
```

```
The following changes will be made to the file's groups:
```

```
Group 1
```

```
age12pewudxq53khcgm49flqq7t6l5na8jscsnn4lqyxla4nzzm4l92qsk7qq4
```

```
--- age1g45d2ymssutc3d3qvsk66qagtwwvpejpf4tz9ve8uej4p7tcu5uq5c8qgn
```

```
2023/11/15 17:01:09 File /Users/schlomoschapiro/Downloads/demo.env synced with new keys
```

## Offline Disaster Recovery Decryption Key for SOPS files

While we use AWS/GCP KMS keys to secure our SOPS files, that renders them inaccessible if we don't have access to the AWS/GCP KMS keys. To provide access to our SOPS files in such a case, we encrypt our SOPS files with an additional AGE key that can be used to decrypt the SOPS files offline.

The following is this additional AGE key used in all our SOPS files. We store it in a sealed envelope and the security posture of our SOPS files relies on the fact that nobody has access to or a copy of this key. Opening this envelope gives access to the key and therefore **requires generating a new AGE key and re-encrypting all SOPS files with it, and storing the new key like this key here in a sealed envelope**.

Mozilla SOPS: <a href="https://github.com/mozilla/sops">github.com/mozilla/sops</a>	AGE: <a href="https://github.com/FiloSottile/age">github.com/FiloSottile/age</a>	This document: Offline Disaster Recovery for SOPS	The HOW-TO: Encrypted Files with SOPS

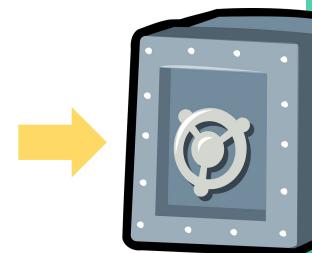
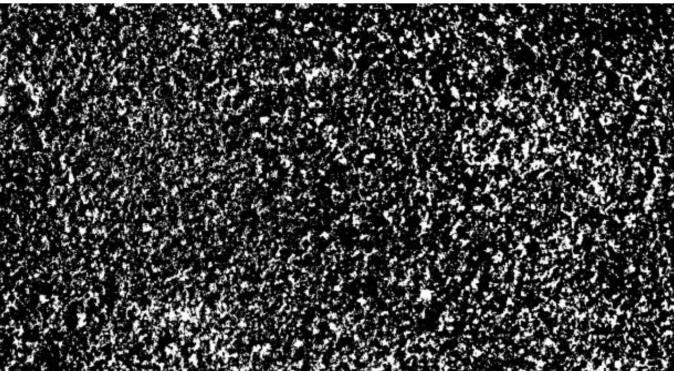
### Usage Hints:

```
SOPS_AGE_KEY=AGE-SECRET... sops -d secrets.yaml # decrypt
SOPS_AGE_KEY=AGE-SECRET... sops -add-age age1... -r -i secrets.yaml # reencrypt
echo AGE-SECRET... | qrencode -s 100 -o key.png # create QR code
```

## Offline Disaster Recovery Decryption Key for SOPS files

CONFIDENTIAL! OPENING THIS REQUIRES RE-ENCRYPTING ALL SOPS FILES! TO BE OPENED BY SRE TEAM!

Key:age1rh03azryv1hmmgecw2v6afar7pvzur767m9cvxp8s9vgzf394gq462kzk



# Fully Automated SOPS Compliance Check

[repo nanny](#)[code search](#)[dependabot PRs](#)[data exports](#)[sign out](#)

▼ SOPS Compliance Found 5 SOPS configuration problems

Found 5 SOPS files with problems, see <a href="https://github.com/tekkit-consulting/tekkit-consulting/pull/123456789">Encrypted

[Line 19 of ./deploy/repo-nanny/secrets.tooling.yaml:](#)

```
sops:
```

No age trust anchors found, add age1n3l6c8ww3ayy6g7w9x75cn4aw0k4v5fxnpnnuymcwgh8euf764vqwruj44

[Line 24 of ./dev-settings.sops.yaml:](#)

```
sops.yaml:24:1:sops:
```

No age trust anchors found, add age1n3l6c8ww3ayy6g7w9x75cn4aw0k4v5fxnpnnuymcwgh8euf764vqwruj44

[Line 12 of ./tests/fixtures/sops-compliance/2\\_old.secrets.yaml:](#)

```
sops-compliance/2_old.secrets.yaml:12:1:sops:
```

No age trust anchors found, add age1n3l6c8ww3ayy6g7w9x75cn4aw0k4v5fxnpnnuymcwgh8euf764vqwruj44

[Line 14 of ./tests/fixtures/sops-compliance/3\\_bad.secrets.json:](#)

```
"sops": {
```

Mandatory age trust anchors (age1n3l6c8ww3ayy6g7w9x75cn4aw0k4v5fxnpnnuymcwgh8euf764vqwruj44) not found



# Cloud & Offline Secrets Management & Disaster Recovery 😊

Managing operational secrets with SOPS

**No Backup?  
No Mercy!**



# Q&A — How may I help you?



[schlomo.schapiro.org](http://schlomo.schapiro.org)

*We are not consultants. We are Partners, Coaches, Humans, Enablers, Catalysts, Sparring Partners, Experts  
... and sometimes a little annoying.*

I focus on IT strategy, IT governance, technology and architecture management, security and compliance automation, related organisational changes, business continuity, open source and cloud technologies – and I'm available as a Principal Engineer or Technical Product Owner for short-term / interim support.

Examples:

- **Business-IT alignment & leveraging**, developing required skills and abilities for 21<sup>st</sup> century IT, leverage AI
- **SaaS compliance & governance**, data possession vs. ownership, IAM, integrations, backup & DR, shadow IT
- **Compliance Automation**, finding the “golden path” to a “golden state”
- **Secrets Management** for Datacenter, Cloud Infrastructure, IaaS/PaaS/SaaS
- **Open Source**, from usage to contribution, writing policies, using SBOM, establishing Open Source Stewardship
- **Good Engineering Practices**, GitOps, test driven development, good architecture decisions, known tech strategy
- **Business Continuity and Disaster Recovery** for office, Cloud infrastructure, data center & SaaS, with quality assurance, emergency communication & collaboration, hot & cold standby, no-restore solution, ransomware protection, Linux Disaster Recovery / Bare Metal Restore with “Relax and Recover ([rear](#))” Open Source tooling

[schlomo.schapiro@tektitconsulting.com](mailto:schlomo.schapiro@tektitconsulting.com)



We ask for  
your feedback!

PLEASE  
VOTE  
NOW!

