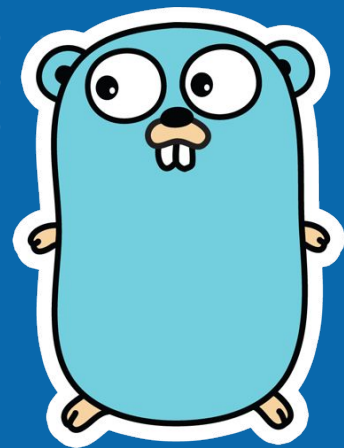


Batch



Meetup Go Lyon
mardi 20 juin

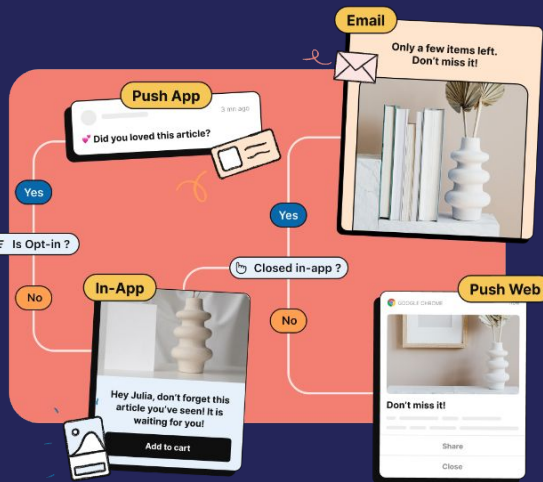
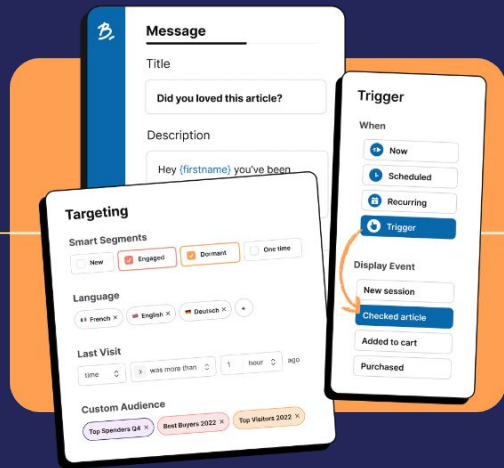
Votre plateforme d'engagement client 3 en 1

La meilleure solution marketing pour construire une relation forte avec vos clients.

VUE COMPLÈTE
DE TOUS VOS CLIENTS

DÉPLOYEZ FACILEMENT LES
WORKFLOWS
DE VOS RÊVES

EXÉCUTEZ VOS SCÉNARIOS SUR
TOUS LES CANAUX



Toutes vos données

Obtenez une vue complète de vos clients, de toutes leurs actions depuis toutes les sources.

The screenshot displays the Silent Corp dashboard. The top navigation bar includes links for Analytics, Userbase, Campaigns, Automations, and Settings. The main navigation tabs are General, Push settings, Team, Custom data, Audiences, Themes, Profiles (selected), Privacy, Labels, and Capping. The 'Profiles' section shows a search bar with 'ancieux' and a 'Search' button. The profile for 'Cécile Ancieux' is displayed, showing details like Location (France), Language (French), Customer ID (lmd_810293), Email (cancieux@832dgql.com), and Phone (+33626364714). Below this, there are tabs for 'Marketing' and 'Transactional'. The 'Live events' section shows a list of events for Monday 30 January 2023, including 'e.cart_ordered', 'e.cart_added', 'e.browsed_category', and 'e.cart_emptied'. The 'Informations' section includes a search bar for attributes, events, and tags, and a table of attributes and events.

Profiles

Q ancieux x Search

Cécile Ancieux

Location: France, Language: French, Customer ID: lmd_810293

Email: cancieux@832dgql.com, Phone: +33626364714

Marketing, Transactional

Push: c-RPzHKHSXaZ-hyPrvV...

Live events

Monday 30 January 2023

- 18:30pm e.cart_ordered Iphone XR See attributes
- 18:17pm e.cart_added Iphone XR See attributes
- 18:13pm e.cart_added Iphone XR See attributes
- 18:11pm e.browsed_category Iphone XR See attributes
- 18:10pm e.cart_added Iphone XR See attributes
- 18:02pm e.browsed_category Iphone XR See attributes
- 17:53pm e.cart_emptied Iphone XR See attributes
- 17:08pm e.cart_added Iphone XR See attributes

Informations

Q Search attributes, events, tags... All properties

Attributes

advertise:	true	adjust_campaign:	"Expired..."
product_code:	PRE	remake_aec_2...	834

Events

cart_added:	14 data	cart_emptied:	6 data
cart_ordered:	6 data	cart_added:	6 data

BÉNÉFICES



Rassemblez vos données depuis **toutes vos sources**



Collectez **100% de votre trafic**, logués ou non logués



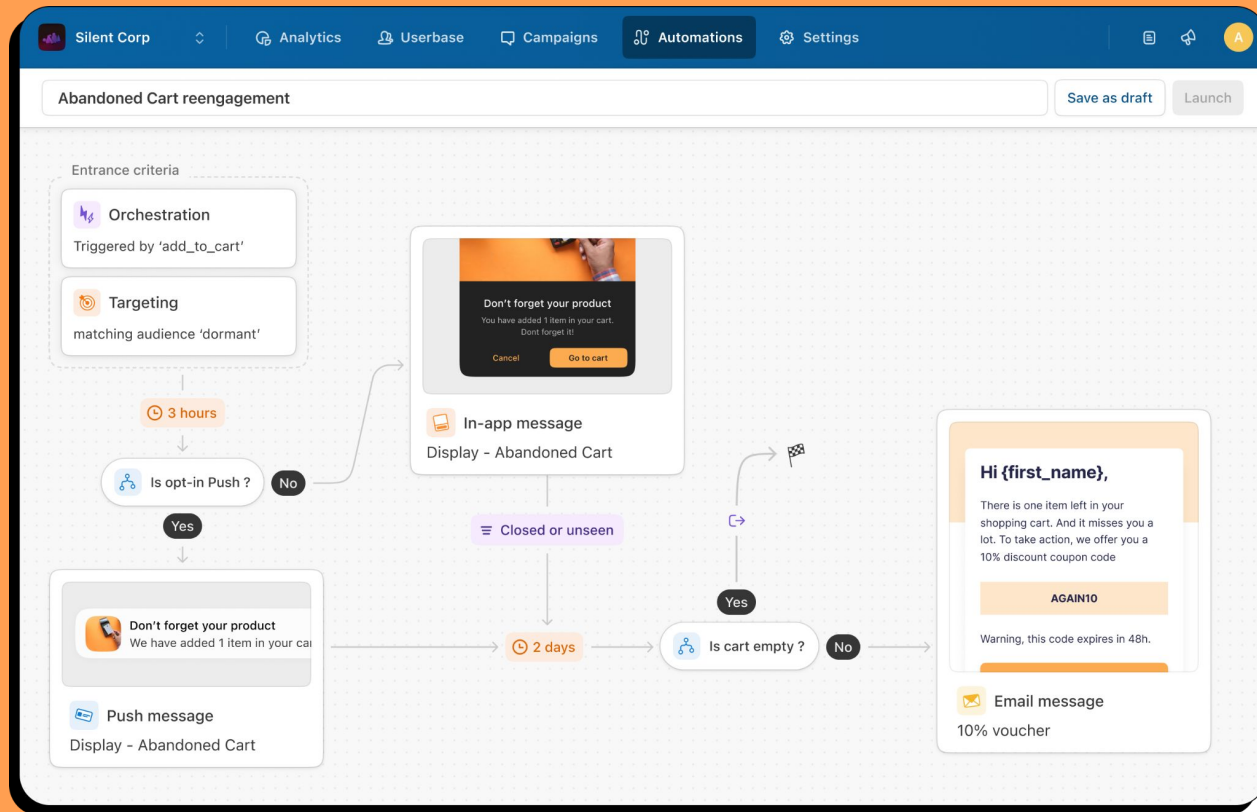
Vue 360° de tous vos clients



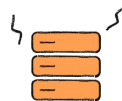
Stockage sécurisé pour la **confidentialité des données**

Tous vos scénarios

Déployez facilement tous vos workflows, depuis un seul et même outil.



BÉNÉFICES



Conçu et utilisable
en **temps réel**



Construisez **tous vos scénarios** avec un seul outil



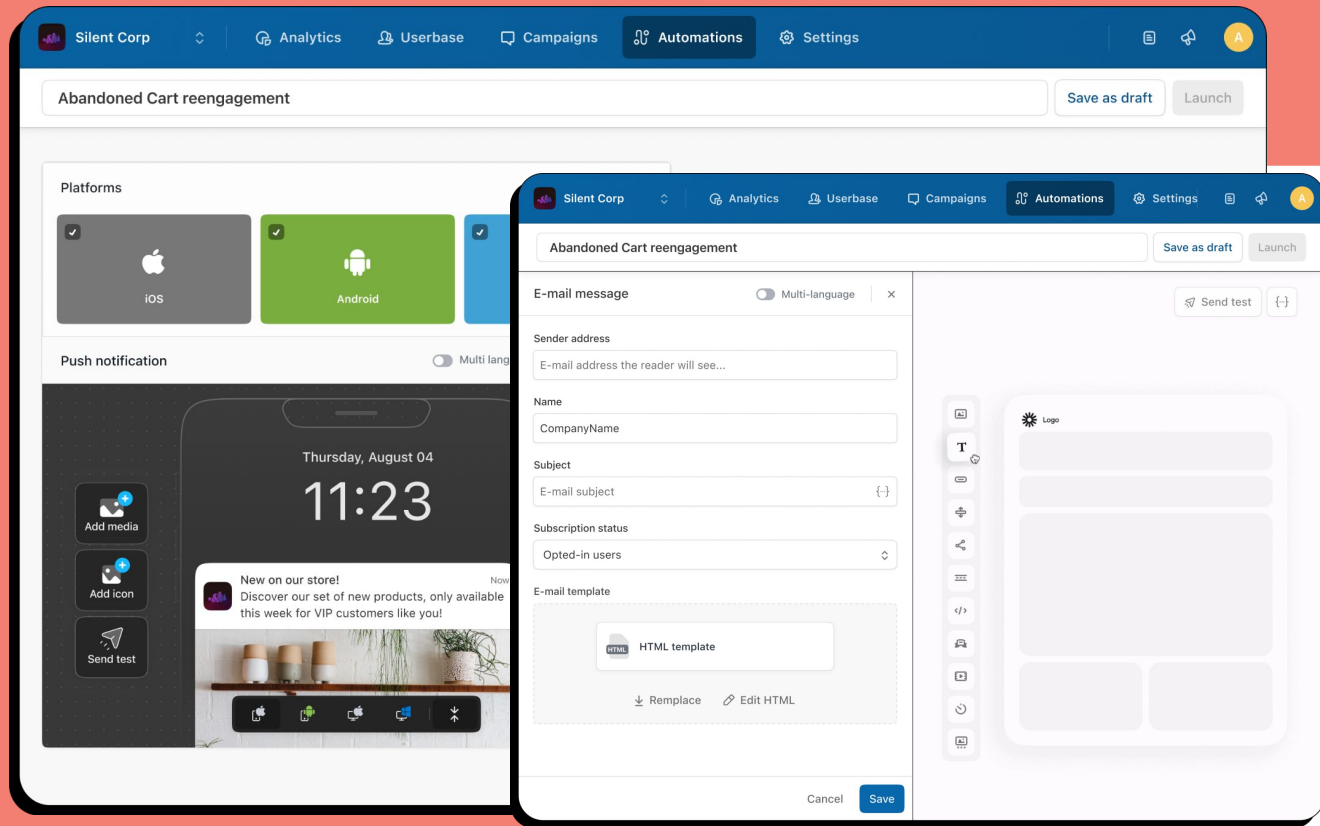
Ciblez facilement avec critères
de sélection **prêts à l'emploi**



Pensé pour améliorer
votre **productivité**

Tous vos canaux

Créez, visualisez & exécutez vos scénarios sur tous vos canaux.



BÉNÉFICES



Envoyez sur **tous vos canaux** depuis la même interface



Créez vos messages en 3 minutes sans développeur



Personnalisation avec toutes les données disponibles



99.9% de taux de délivrabilité sur 650 Mrds de messages

À propos de *Batch*

La plateforme d'engagement client

+8

Années d'expérience



350

Clients



100

Employés



Enterprise

Ready



GDPR Compliant

99.9% Uptime

Connectors & Integrations

sncfconnect

leboncoin

Le Monde

rnd RedaktionsNetzwerk Deutschland

EUROSPORT

aufeminin

DECATHLON

l'express

webedia/.

PM
PRISMA MEDIA

castorama

L'ÉQUIPE

radiofrance

FNAC DARTY

sky

BVG

EDF

Edenred

DER TAGESSPIEGEL

AXA

SOCIÉTÉ GÉNÉRALE

DR

FDJ

BNP PARIBAS
La banque d'un monde qui change

beIN
SPORTS

RATP

pass
Culture

SEPHORA

Rakuten

TF1

France

MICHELIN

showroomprive.com
Responsable de votre plaisir

Schneider
Electric

Cityscoot

MONOPRIX

FFF

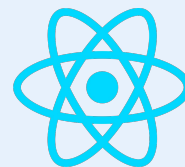
ManoMano

ACCOR

L'ORÉAL

notre stack

- Platform : Golang, Java/Kotlin, Cassandra, Kafka, Protobuf, gRPC.
- Data : Python, Airflow, Flink.
- SRE/DevOps : Ansible, Kubernetes, HAProxy, Grafana, Prometheus.



React



kafka

la plateforme

Batch en chiffres

- **10K+ apps et sites web** utilisent notre SDK/tag en prod,
- 1500 users/mois sur notre dashboard web,
- **500 milliards de notifications** délivrées chaque année
- **50B+ d'events par mois** en entrée sur nos web services
- **400 serveurs** (300x bare-metal, 100x cloud instances)

Batch

Merci



hey 🙌,
on recrute :)

herve@batch.com

vincent@batch.com

Batch

**Import paths custom
repo privés
proxy de modules**

De quoi on parle ?

- Les "import paths" des modules Go sont en réalité des URL
- On utilise souvent github.com ou gitlab.com par défaut car c'est là qu'on héberge le code
- On peut cependant utiliser un domaine custom comme go.foobar.com

Import path standard:

- `github.com/vrischmann/envconfig`
- `github.com/spf13/cobra`

Import path custom:

- `go.uber.org/zap`
- `go.opentelemetry.io/otel`
- `golang.b47ch.com/typedid/v2`

Batch

**Utiliser un domaine
custom**

Fonctionnement de la commande Go

La commande Go a une connaissance interne de certains sites d'hébergement de code comme GitHub

1. Match automatique sur les import github.com
2. Téléchargement + parsing de l'import path pour chercher un *meta tag*

```
// vcsPaths defines the meaning of import paths referring to
// commonly-used VCS hosting sites (github.com/user/dir)
var vcsPaths = []*vcsPath{
    // GitHub
    {
        pathPrefix: "github.com",
        regexp:     lazyregexp.New(`^(?P<root>github\.com/[\\w.-]+/[\\w.-]+)(/[\\w.-]+)*$`),
        vcs:         "git",
        repo:        "https://{root}",
        check:       noVCSuffix,
    },

    // Bitbucket
    {
        pathPrefix: "bitbucket.org",
        regexp:     lazyregexp.New(`^(?P<root>bitbucket\.org/(?P<bitname>[\\w.-]+/[\\w.-]+))(/[\\w.-]+)*$`),
        vcs:         "git",
        repo:        "https://{root}",
        check:       noVCSuffix,
    },
}
```

```
// repoRootForImportDynamic finds a *RepoRoot for a custom domain that's not
// statically known by repoRootFromVCSPaths.
//
// This handles custom import paths like "name.tld/pkg/foo" or just "name.tld".
func repoRootForImportDynamic(importPath string, mod ModuleMode, security web.SecurityMode) *RepoRoot {
    url := urlForImportPath(importPath)

    resp := web.Get(security, url)
    body := resp.Body

    imports := parseMetaGoImports(body, mod)

    // Find the matched meta import.
    mmi := matchGoImport(imports, importPath)

    validateRepoRoot(mmi.RepoRoot)
    vcs := vcsByCmd(mmi.VCS)

    CheckGOVCS(vcs, mmi.Prefix)

    repoURL, ok := interceptVCS(mmi.RepoRoot, vcs, security)
    if !ok {
        repoURL = mmi.RepoRoot
    }

    rr := &RepoRoot{
        Repo:  repoURL,
        Root:  mmi.Prefix,
        IsCustom: true,
        VCS:   vcs,
    }

    return rr
}
```

Domaine custom

Avec ce qu'on a vu, on voit qu'il suffit de fournir ce fameux *meta tag* et Go saura retrouver le repo qui se cache derrière.

N'importe quel serveur capable de servir du HTML statique fera l'affaire.

```
$ curl https://go.uber.org/zap
<!DOCTYPE html>
<html>
<head>
<meta name="go-import" content="
    go.uber.org/zap
    git
    https://github.com/uber-go/zap
">
</head>
</html>
```

C'est quoi ce proxy ?

Par défaut, tous les téléchargements de modules Go passent d'abord par `proxy.golang.org`.

Pratique pour les modules publics mais logiquement ne connaît pas nos modules privés.

Description de `proxy.golang.org`

The Go team is providing the following services run by Google: a module mirror for accelerating Go module downloads, an index for discovering new modules, and a global `go.sum` database for authenticating module content.

Since Go 1.13, the `go` command by default downloads and authenticates modules using the Go module mirror and Go checksum database

Plusieurs solutions pour ignorer le proxy

1. Utiliser
GOPRIVATE=*.example.com
2. Utiliser
GONOPROXY=*.example.com et
GONOSUMDB=*.example.com

Description de `go help environment`

GOPRIVATE

Comma-separated list of glob patterns (in the syntax of Go's `path.Match`) of module path prefixes that should always be fetched directly or that should not be compared against the checksum database.

Batch

**Ok, mais si je veux mon
propre proxy ?**

Le protocole de GOPROXY

- La spec est ouverte et assez simple: 5 routes à implémenter
- Peut être servi avec un simple serveur de fichiers statique
- En pratique on veut que ça aille fetch les modules depuis GitHub en fallback

```
/golang.b47ch.com/typedid/v2/@v/list  
/golang.b47ch.com/typedid/v2/@latest  
/golang.b47ch.com/typedid/v2/@v/v2.0.0.info  
/golang.b47ch.com/typedid/v2/@v/v2.0.0.mod  
/golang.b47ch.com/typedid/v2/@v/v2.0.0.zip  
/golang.b47ch.com/typedid/v2/@v/v2.2.0.info  
/golang.b47ch.com/typedid/v2/@v/v2.2.0.mod  
/golang.b47ch.com/typedid/v2/@v/v2.2.0.zip
```

Implémentation du proxy

- Il existe plein de solutions pour avoir son propre proxy, on a choisi d'intégrer une lib qui fournit un simple handler HTTP
- Moins de 100 lignes de code pour avoir un proxy fonctionnel
- À activer avec la variable GOPROXY

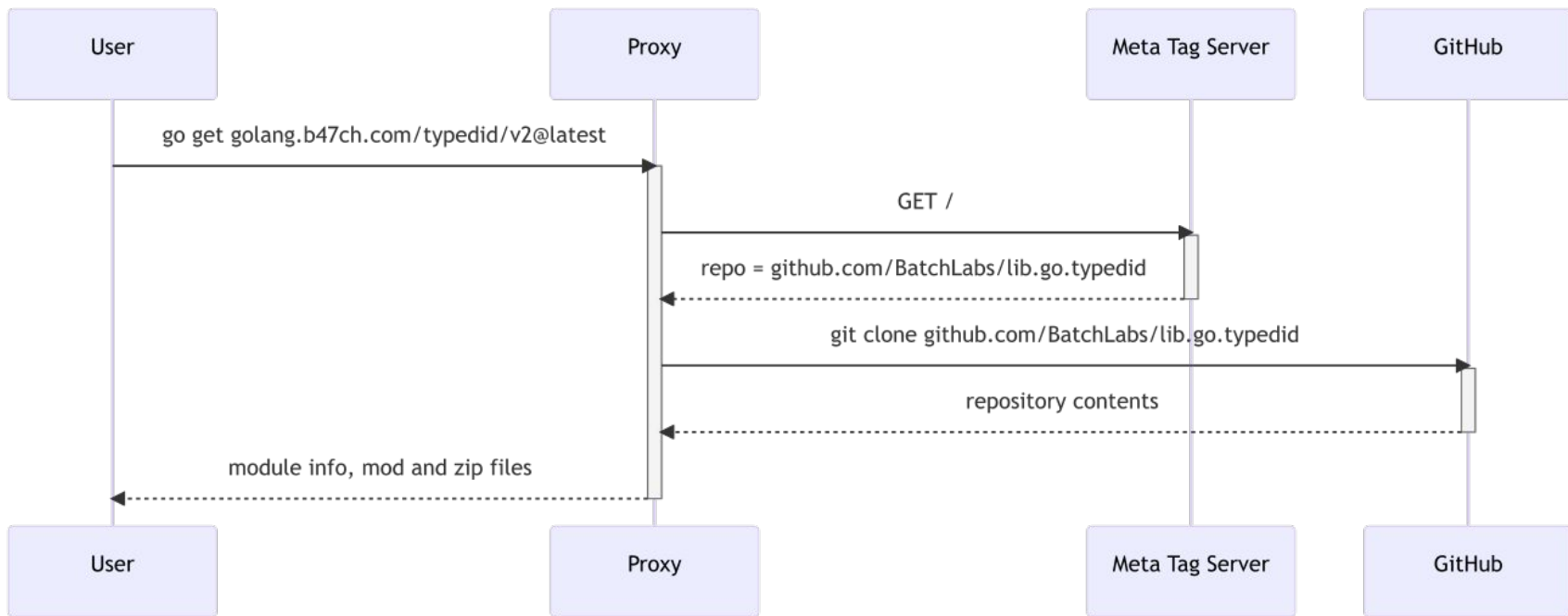
```
proxy := &goproxy.Goproxy{
    GoBinMaxWorkers: 4,

    GoBinEnv: []string{
        "PATH=/usr/bin",
        "HOME=" + os.Getenv("HOME"),
        "GOPRIVATE=*.b47ch.com",
    },

    Cacher: goproxy.DirCacher("/data/goproxy"),
}

http.ListenAndServe(":8080", proxy)
```

Mis bout à bout



Batch

Quid des modules privés ?

Easy Mode

On peut utiliser un *Personal Access Token* ou PAT. Il faut simplement configurer le fichier *.netrc*.

Avantages:

- Simple à mettre en place
- Permet un contrôle fin des repos accessibles

Inconvénients:


- Besoin de changer le PAT régulièrement
- Pas très bien fait, gros flou sur qui est le vrai propriétaire d'un PAT: la personne ou l'organisation GitHub ?

```
$ cat ~/.netrc
machine github.com
login vrischmann
password github_pat_ABCXYZ123
```

Hard Mode

GitHub permet de faire des *GitHub Apps*. Une app utilise une clé privée RSA pour générer un JWT et s'authentifie avec ça.

Mais, besoin de quelques composants supplémentaires...

 **Go Modules Proxy**
🕒 Installed last year 👤 Developed by [BatchLabs](#) ⚙️ [App settings](#) 🔗 <https://batch.com/>

Application nécessaire pour accéder à nos repos Go depuis les actions et les builders docker.


Permissions

✓ **Read** access to code and metadata





Repository access

☐ **All repositories**
This applies to all current and future repositories owned by the resource owner.
Also includes public repositories (read-only).

☒ **Only select repositories**
Select at least one repository.
Also includes public repositories (read-only).

 **Select repositories** ▾

Selected 38 repositories.

 BatchLabs/ lib.go.backoff	×
 BatchLabs/ lib.go.gocql	×
 BatchLabs/ lib.go.buzz	×
 BatchLabs/ lib.go.sender	×

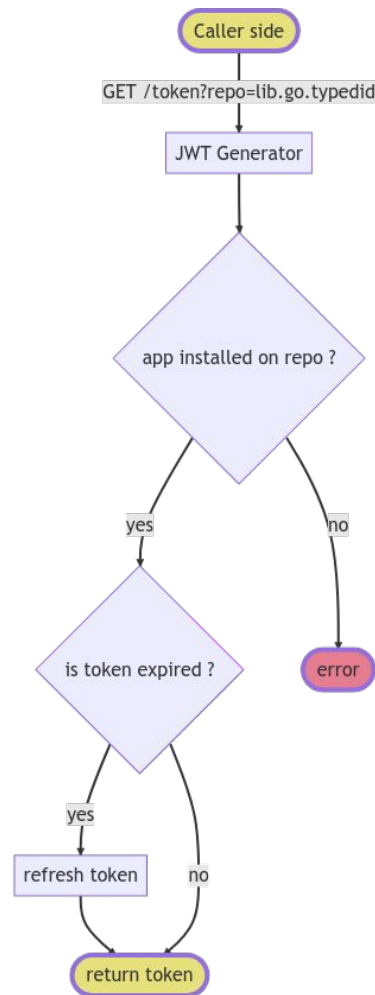
git credential helper

- définit une commande à exécuter pour obtenir les credentials d'un repo particulier
- notre helper va interroger un *serveur de JWT* en HTTP

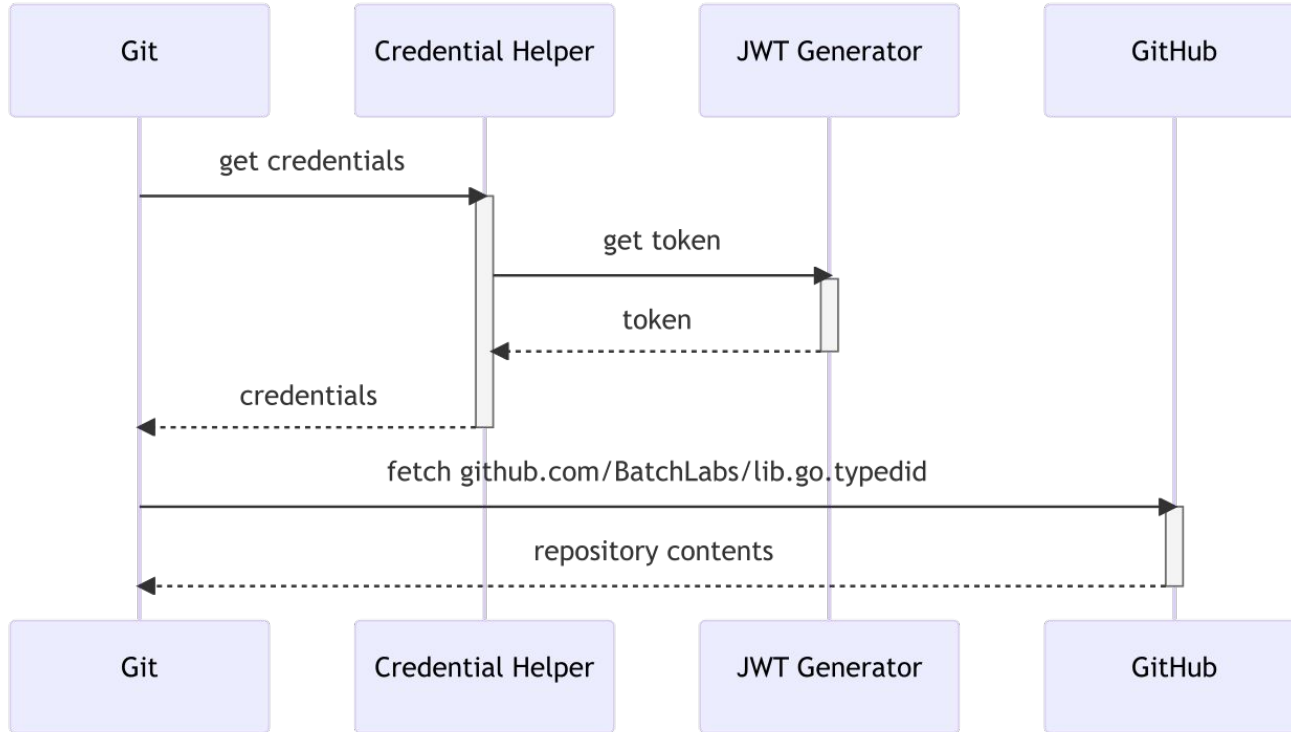
```
[credential "https://github.com/BatchLabs/"]  
  helper = !git-credential-helper  
  useHttpPath = true
```


JWT Generator

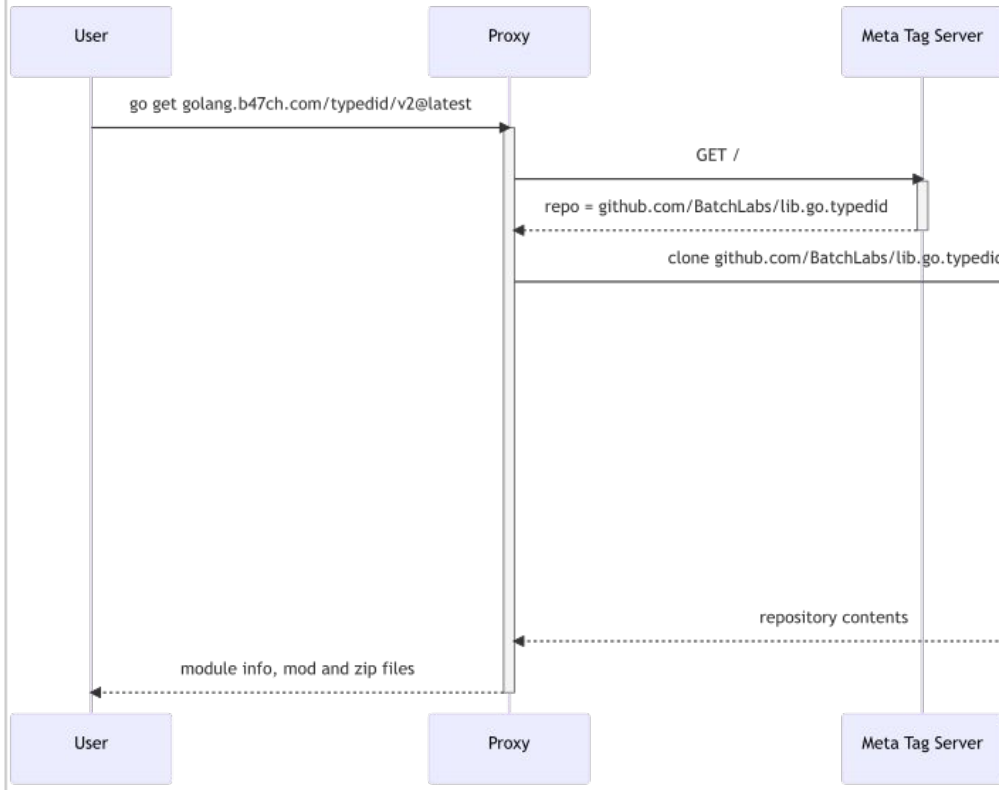
- Serveur HTTP capable de s'authentifier auprès de GitHub en tant qu'installation d'une app GitHub
- Reçoit une requête pour un repo donné, valide que l'app est bien installé sur le repo
- Génère un JWT et le renvoie en réponse
- Purement local, ne sert qu'à faire de l'IPC avec le *git credential helper*



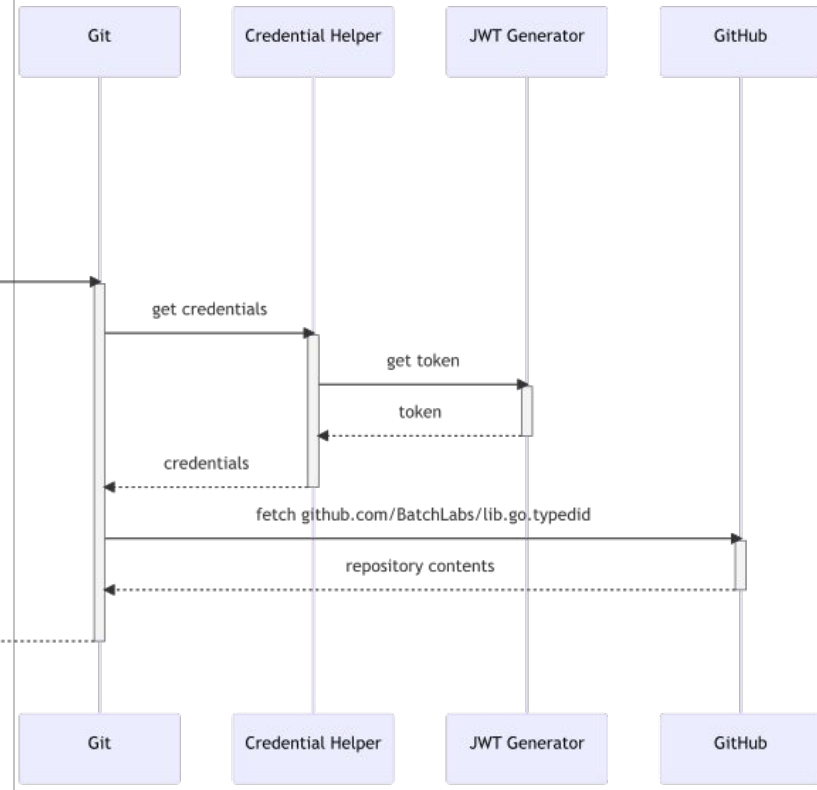
Mis bout à bout



Request module from proxy



Request private repo content from GitHub



Batch

Fin !

Questions ?