

DIRAC et le langage Go

Évaluation du langage de programmation Go pour DIRAC

Christophe Meessen (CPPM)

Objectifs

- Réaliser un client ou serveur DIRAC en Go
- Interactions avec clients ou serveurs DIRAC en python

Pourquoi Go ?



- Langage simple
- Compilation rapide dans un seul fichier exécutable
- Ramasse miette intégré (*Garbage Collector*)
- Parallélisme intégré dans le langage avec détection de *deadlocks* et de *race condition*
- Sécurité avec contrôle des limites pour les tableaux
- Http, crypto et SQL dans la bibliothèque standard
- Support de tests et mesure de couverture intégrés

Réalisations (1/2)

- Configuration
 - Lire les données de configuration d'une instance de DIRAC
 - Configuration dans fichiers, en ligne de commande, ou de serveurs (http)
 - Rafraîchissement périodique des données des serveurs

```
import "config"
```

```
val, ok := config.ValueString("/DIRAC/...", "default value")
```

```
x, _ := config.ValueInt("/DIRAC/...", 123)
```

```
z, _ := config.ValueSlice("/DIRAC/...", []string{"a", "b"})
```

Réalisations (2/2)

- AuthManager
 - *Query(method, credential) bool*
 - Renvoie True si credential autorise l'exécution de method, False dans le cas contraire

```
import "auth"

// instantiate an auth.Manager
mgr := auth.New("/Framework/.../Services/Monitoring/Authorization")
if mgr.Query("tryView", credentials) {
    // allowed to invoke tryView
}
```

À faire

- Logging avec logLevel et multiple backend
- Protocole de communication : HTTPS et JSON
 - Pas utiliser DISET et DEncode

Difficultés

- HTTPS non disponible pour tester l'interopérabilité
- JSON disponible, mais non utilisé
- Pas d'exemples de requêtes/réponses en JSON
- JEncode encode des objets python (classe & module), objets non compatible avec Go

Conclusion

- Ne garder que l'essentiel simplifie la base de code
- Seules limitations techniques avec Go :
 - Pas de chargement dynamique de code (tout est compilé)
 - Application monolithique avec des micro services (pas plugin)
 - Transmission d'objets JSON implique code prédéfini dans Go
- Aucune autre limitation à signaler
- L'interopérabilité Python-Go est le plus gros challenge