

Distributed Systems, 3rd edition,
MvS&AST,
Architectures

Międzymordzia (interfejsy)

Opracował Maksymilian Zawartko

Usługa nie staje się prostsza ani trudniejsza przez oferowane przez nią międzymordzie. Usługa oferuje funkcjonalność, a międzymordzie definiuje tylko sposób korzystania z niej. Problem wyboru między międzymordziem REST(ful), a specyficznym dla usługi, można sprowadzić do przezroczystości dostępu (access transparency).

Amazon S3 (usługa przechowywania danych w chmurze)

oferuje zarówno międzymordzie REST, jak i tzw. SOAP:

Operacje na komorach (buckets)

- ListAllMyBuckets
- CreateBucket
- DeleteBucket
- ListBucket
- GetBucketAccessControlPolicy
- SetBucketAccessControlPolicy
- GetBucketLoggingStatus
- SetBucketLoggingStatus

Operacje na obiektach

- PutObjectInline
- PutObject
- CopyObject
- GetObject
- GetObjectExtended
- DeleteObject
- GetObjectAccessControlPolicy
- SetObjectAccessControlPolicy

Pythonowa biblioteka **boto**

oferuje jeszcze więcej, bo ok. 50 operacji dostępu do usługi Amazon S3. Tymczasem międzymordzie REST - tylko 4.

- **PUT** - modyfikuje dany zasób przesyłając jego nowy stan
- **POST** - tworzy nowy zasób
- **GET** - pobiera dany zasób
- **DELETE** - usuwa dany zasób

Wynika to z faktu, że w architekturach REST(ful) aplikacja musi dokładnie opisać swoje żądanie parametrami przekazywanymi do jednej z operacji REST. W międzymordziu SOAP liczba parametrów jest bardziej ograniczona, a w przypadku biblioteki **boto** - jeszcze bardziej.

Niech **bucket** będzie hipotetycznym międzymordziem, oferującym operację **create**, która jako argument przyjmuje napis, będący nazwą tworzonej komory.

```
import bucket
```

```
bucket.create("mybucket")
```

W architekturze REST(ful) wszystkie informacje potrzebne do wykonania operacji trzeba zawrzeć w jednym napisie. Mogłoby to wyglądać np. tak.

```
PUT "http://mybucket.s3.amazonsws.com/"
```

W przypadku specyficznego międzymordzia wiele błędów składniowych może zostać odkrytych już przy kompilacji, natomiast w architekturze REST(ful) - dopiero podczas wykonania programu (runtime). Poza tym użycie dedykowanych międzymordzi jest prostsze, niż tych ogólnych. Z drugiej jednak strony użycie generycznych operacji ułatwia dokonywanie zmian - wystarczy zmienić napis (argument).