# Mini-Zanzibar: Globalni Sistem Autorizacije

Studenti treba da implementiraju pojednostavljenu verziju Zanzibara (u daljem tekstu mini Zanzibar), globalnog sistema autorizacije kojeg je definisao Google. Kratko objašnjenje i demonstraciju možete videti na web stranici, dok šire objašnjenje možete naći u objavljenom naučnom radu.

Ideja je da na projektu primenite znanje o razvoju bezbednog softvera stečeno kroz dosadašnje zadatke. Uz to, imaćete prilike da naučite više o kontroli pristupa, konzistentnosti, skalabilnosti.

Vaša verzija sistema bi trebalo da:

1. Podržava fleksibilan konfiguracioni jezik za definisanje politika kontrole pristupa.
2. Skladišti i evaluira liste kontrole pristupa (ACLs).
3. Obezbeđuje konzistentne i skalabilne odluke o autorizaciji.
4. Postiže nisko kašnjenje (*latency*) i visoku dostupnost (*availability*) za proveru autorizacije.

**1. Model Podataka**

Kako bi vam bilo jasnije kako bi trebalo da izgleda model, obavezno pogledajte demonstraciju ("*See how it works*" sekcija njihove web stranice), pri čemu su dovoljni samo prvi i drugi korak ("Basics" i "Editors -> Viewers").

- **Relacione Torke**: Skladištite ACL-ove kao relacione torke koje povezuju objekte sa korisnicima sa specifičnim relacijama. Za skladištenje ovog tipa podataka koristite *Google*-ov LEVEL DB.
    - **Format**: **object#relation@user**
    - **Primer**: **doc:readme#viewer@user:alice**

**2. Konfiguracija namespace-a:**

- **Namespace**: Konfiguracija koja definiše relacije i parametre skladištenja. Definišite različite tipove pristupa, npr. **owner**, **editor**, **viewer,** kao i način na koji se računaju relacije između korisnika i objekata. Za skladištenje ovoga koristite ConsulDB sa verzionisanjem.
    - **Relacije**: Definišite različite tipove pristupa, npr. **owner**, **editor**, **viewer**. Potrebno je omogućiti i definisanje koncentričnih relacija.
    - **Pravila prepisivanja skupa korisnika**: Definišite kako se relacije između korisnika i objekata računaju. U okviru pravila je dovoljno definisati samo operaciju *union.*
    - *Tuple_to_userset* funkciju **nije** potrebno implementirati!
    - Grupe korisnika (*userset*) i vezanu aritmetiku **nije** potrebno implementirati.
    - **Primer konfiguracije**:

```
{
    "namespace": "doc",
    "relations": {
        "owner": {},
        "editor": {
            "union": [
```

```
                {"this": {}},
                {"computed_userset": {"relation": "owner"}}
            ]
        },
        "viewer": {
            "union": [
                {"this": {}},
                {"computed_userset": {"relation": "editor"}}
            ]
        }
    }
}
```

**4. API**

- **Kreiranje/Izmena ACL-a**:
    - Endpoint: **POST /acl**

```
{
    "object": "doc:readme",
    "relation": "viewer",
    "user": "user:alice"
}
```

- **Provera ACL-a**:
    - Endpoint: **GET /acl/check**
    - Parametri Zahteva: **object**, **relation**, **user**

```
{
    "authorized": true
}
```

- **Kreiranje/Izmena namespace-ova**
    - Endpoint: **POST /namespace**

**Što se tiče ciklusa razvoja sistema čiji je mini Zanzibar deo, potrebno je:**

- Osmisliti celovit sistem koji se oslanja na mini Zanzibar (pri čemu se samo mini Zanzibar i jedan *proof-of-concept* klijent za test (kojeg ne morate razmatrati u dubinu) moraju biti implementirani).
- Analizirati osmišljeni sistem i definisati bezbednosne zahteve (pogledajte OWASP Application Security Verification Standard, razmislite šta je moguće primeniti u datom kontekstu).
- Definisati kontekstni model pretnji (*threat*) za osmišljeni sistem i procesne modele pretnji vezane za operacije u kojima učestvuje mini Zanzibar.
- Implementirati mini Zanzibar kako je objašnjeno gore u tekstu. Sama implementacija ne mora biti šira od onoga što je traženo (kako ne biste potrošili previše vremena na nju). Ako se nađete u situaciji da vam nije jasno da li bi nešto trebalo implementirati, obavezno pitajte.
- Ispratiti razvoj redovnim komitovima. Uz to, potrebno je da jedno drugima radite (*secure*) *code review* (koristite mehanizam *pull request*-ova).
- Skenirati kreirani softver alatom za statičku analizu koda (predlog je: **Sonar sa Security plugin-om** ili **Github CodeQL).**