

Leasing

(Resource management)

Názvosloví

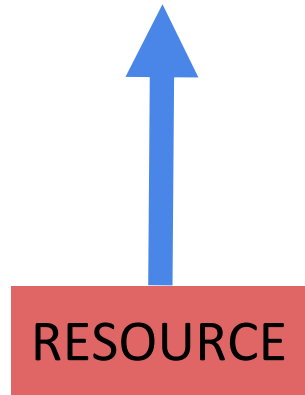
- Resource = Zdroj
- Lease = Pronájem

Leasing - Pojmy

Někdo pronajímá něco někomu na nějakou dobu.

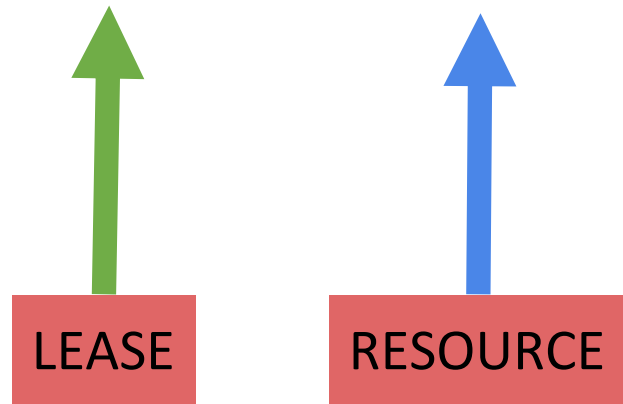
Leasing - Pojmy

Někdo pronajímá **něco** někomu na nějakou dobu.



Leasing - Pojmy

Někdo pronajímá něco někomu na nějakou dobu.



Leasing - Pojmy

Někdo pronajímá něco někomu na nějakou dobu.



GRANTOR



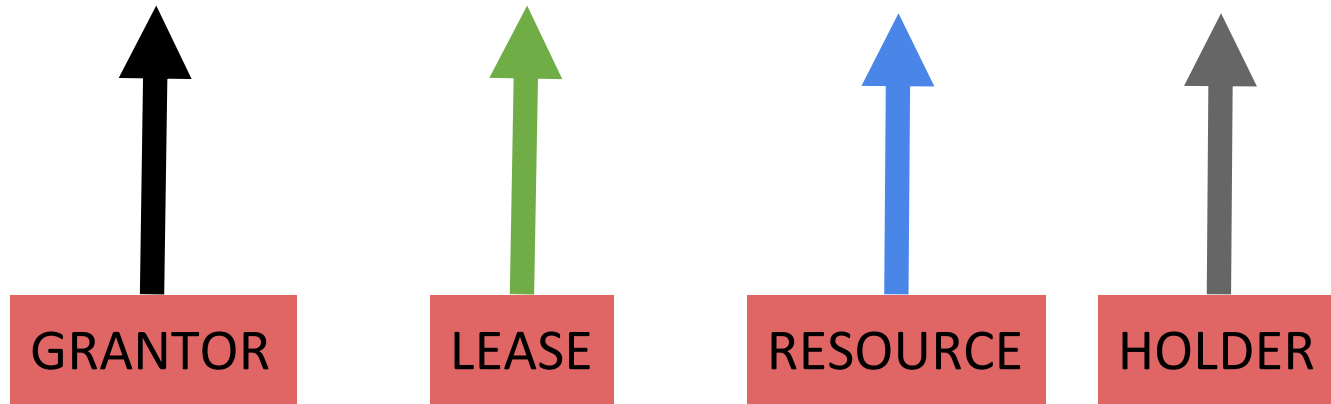
LEASE



RESOURCE

Leasing - Pojmy

Někdo pronajímá něco někomu na nějakou dobu.

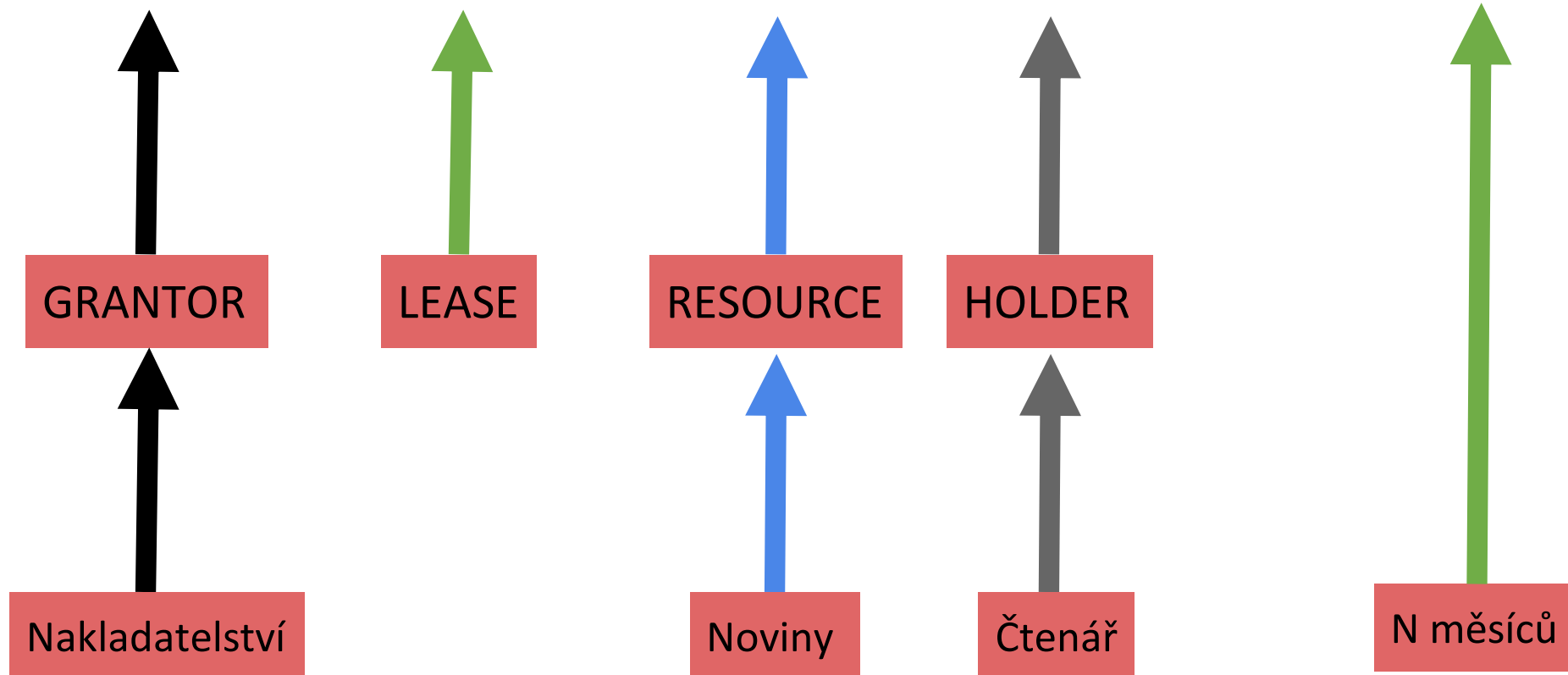


Struktura leasingu

- **Resource (zdroj)** poskytuje funkcionalitu, či službu.
- **Lease (pronájem)** udává čas, po který je zdroj přístupný.
- **Grantor** garantuje pronájem zdroje.
- **Holder** pracuje se zdrojem na dobu omezenou pronájmem.

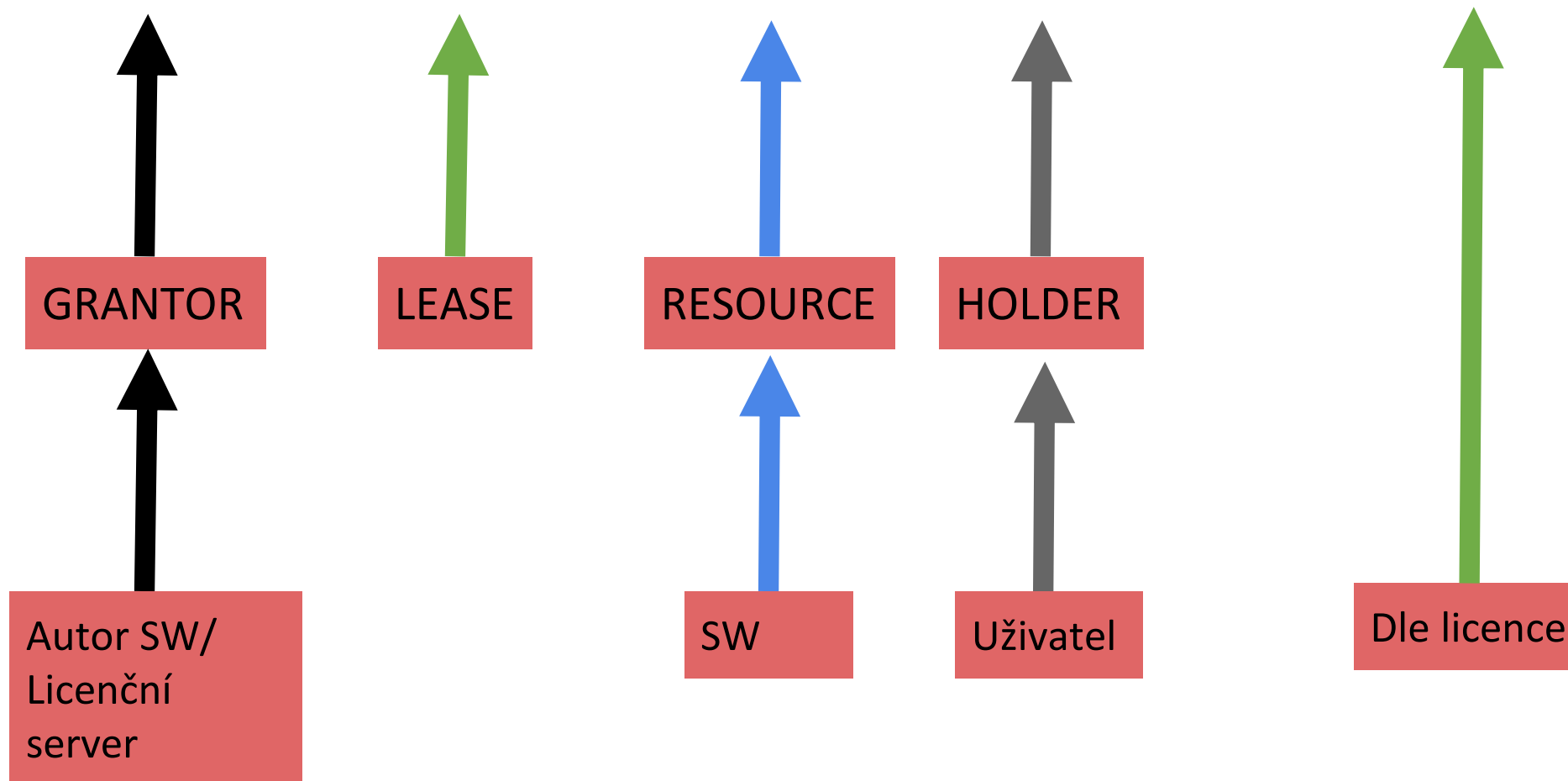
Leasing - Příklad - Novinové předplatné

Někdo pronajímá něco někomu na nějakou dobu.



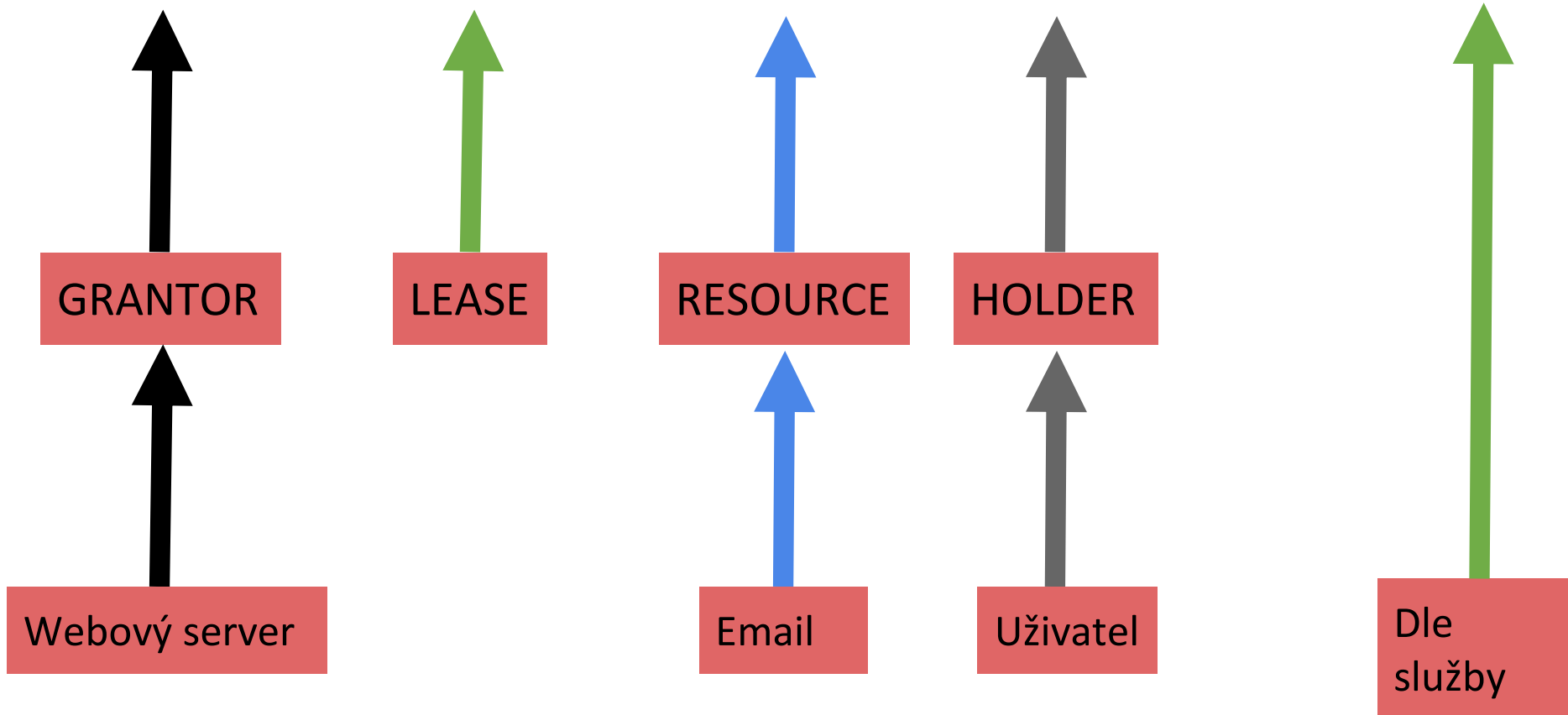
Leasing - Příklad - Softwarové licence

Někdo pronajímá něco někomu na nějakou dobu.



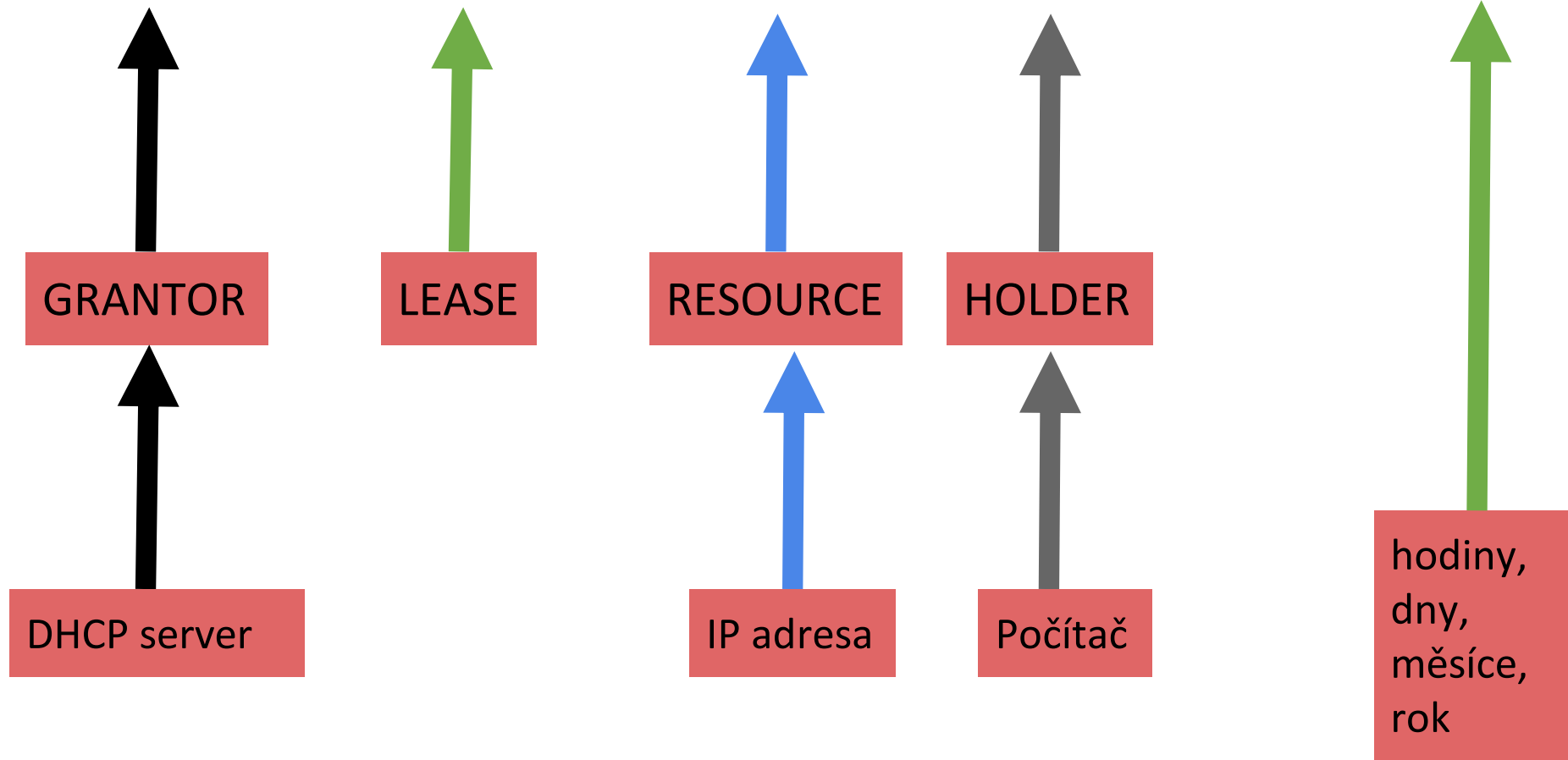
Leasing - Příklad - Web-based emailové účty

Někdo pronajímá něco někomu na nějakou dobu.



Leasing - Příklad - DHCP

Někdo pronajímá něco někomu na nějakou dobu.



DHCP (Dynamic Host Control Protocol)

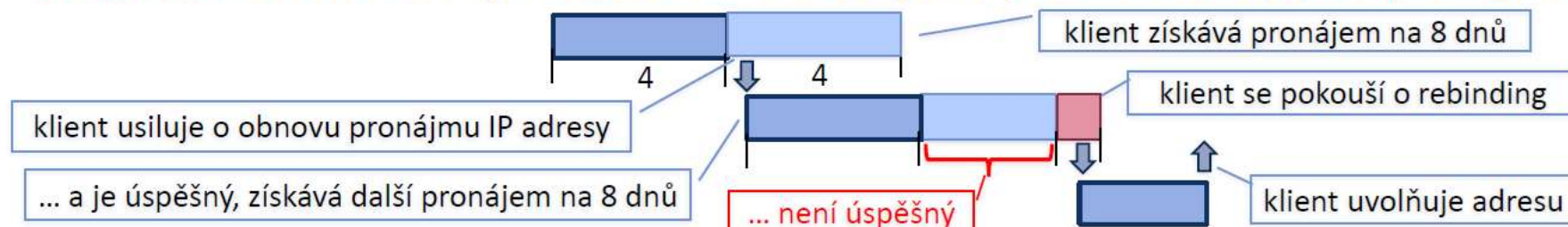
- Vznikl z BOOTP - starší protokol - konfigurace bezdiskových stanic
- Statické (BOOTP) vs dynamické (DHCP) přidělování adresy
- Přidělování adres v DHCP - ručně, trvale, dočasně

- původně:
 - uzel má svou IP adresu přidělenou (ve smyslu: je jejím držitelem, vlastníkem)
 - nemusí se starat o obnovu/prodloužení
 - nevýhodou je malá hospodaření s IP adresami
- při dočasném přidělení (dynamic allocation):
 - uzel má svou IP adresu pouze dočasně pronajatu (leased)
 - a musí se aktivně starat (alespoň) o její obnovu
 - výhodou je větší pružnost při práci s IP adresami
- otázka:
 - na jak dlouho (dočasně) přidělovat IP adresy?
 - kratší doba = větší pružnost
 - delší doba = větší „stabilita“ a větší režie
 - delší doba = méně operací spojené s přidělováním a obnovou
 - delší doba = větší pružnost
 - delší doba = větší „stabilita“ a menší režie
- doba „pronájmu“ (DHCP lease)
 - může být například:
 - hodina
 - den
 - 3 dny (používá Microsoft)
 - týden
 - měsíc
 - rok (skoro už „trvalé“)

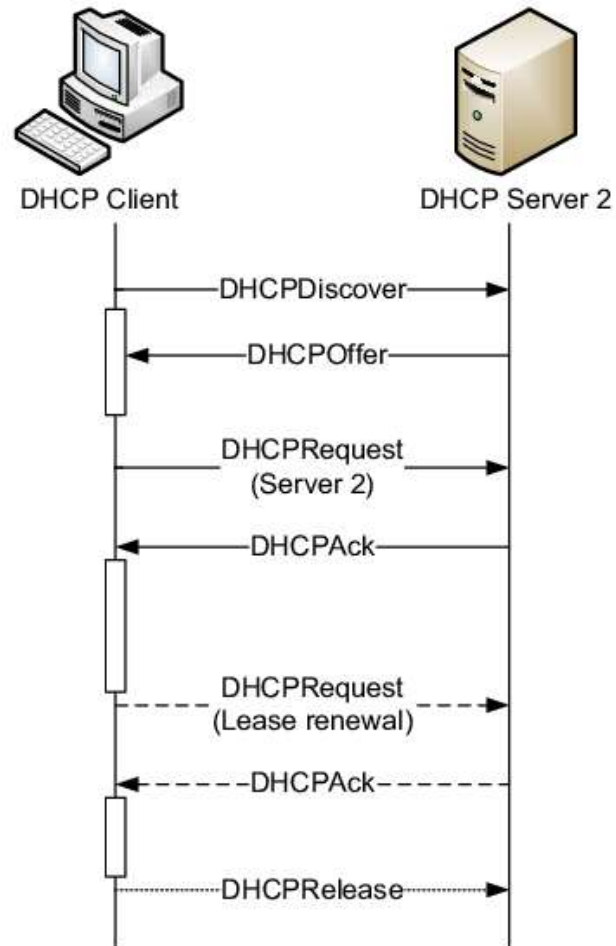
ne přednášku o DHCP - jen stručně princip

„pronájem“

- DHCP klient prochází různými stavy a provádí různé úkony:
 - **alokace**: klient ještě nemá IP adresu a žádá DHCP server o „pronájem“ IP adresy
 - žádá o DHCP lease
 - **realokace**: klient již má „pronajatou“ svou IP adresu, ale snaží si tento pronájem potvrdit
 - například poté, co prošel rebootem či byl na nějakou dobu vypnut
 - ale ještě trvá doba předchozího „pronájmu“
 - **obnova**: před koncem „pronájmu“ se klient snaží o jeho prodloužení
 - kontaktuje DHCP server se žádostí o prodloužení „pronájmu“
 - začíná to zkoušet od poloviny stávajícího pronájmu (timer T1 = 50% lease time)
 - **rebinding**: snaží se získat pronájem stejné IP adresy od jiného DHCP serveru
 - pokud se nepodaří obnova (např. když je původní DHCP server nedostupný), snaží se uzel o získání stejné IP adresy od jiného serveru (timer T2 = 87,5% l.t.)
 - **uvolnění**: klient „vrací“ pronajatou IP adresu ještě před koncem jejího pronájmu

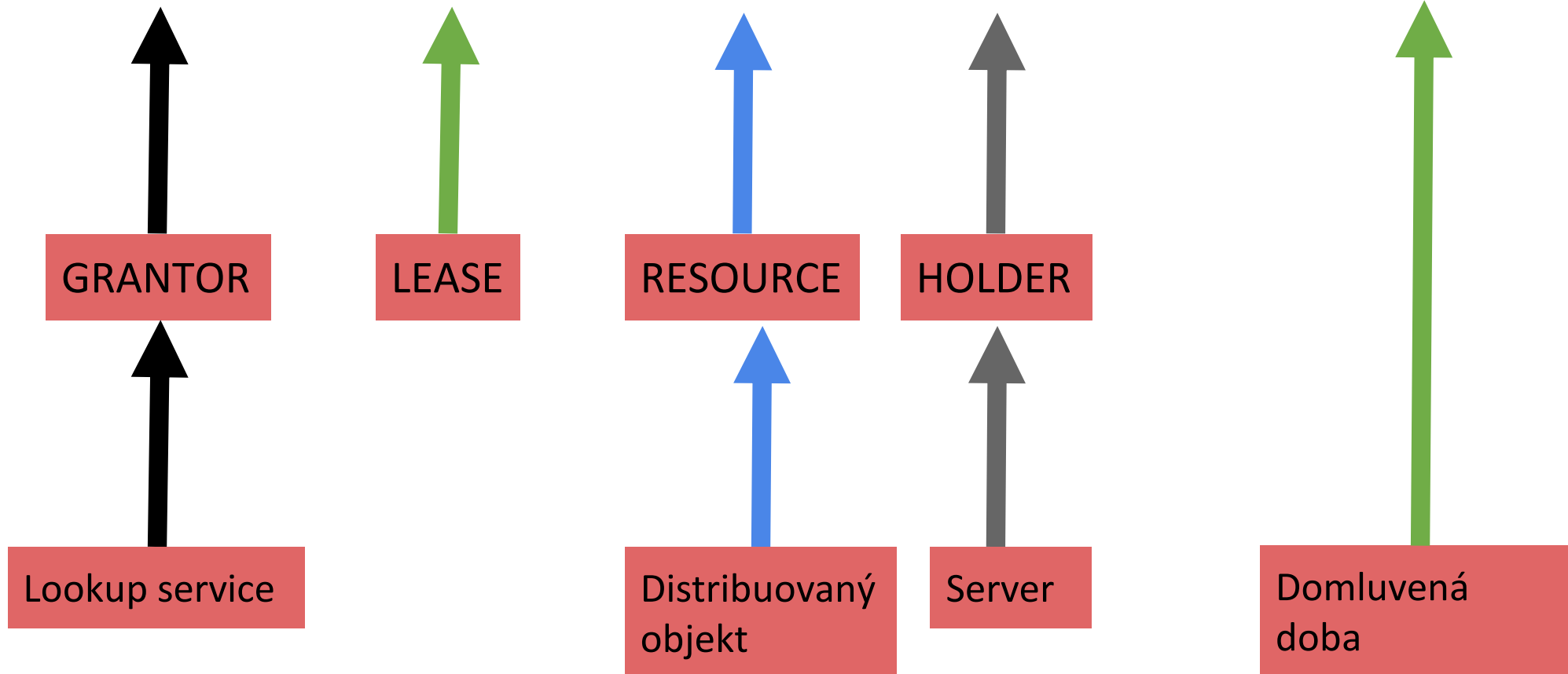


Příklad: DHCP (Dynamic Host Configuration Protocol)



Leasing - Příklad - CORBA

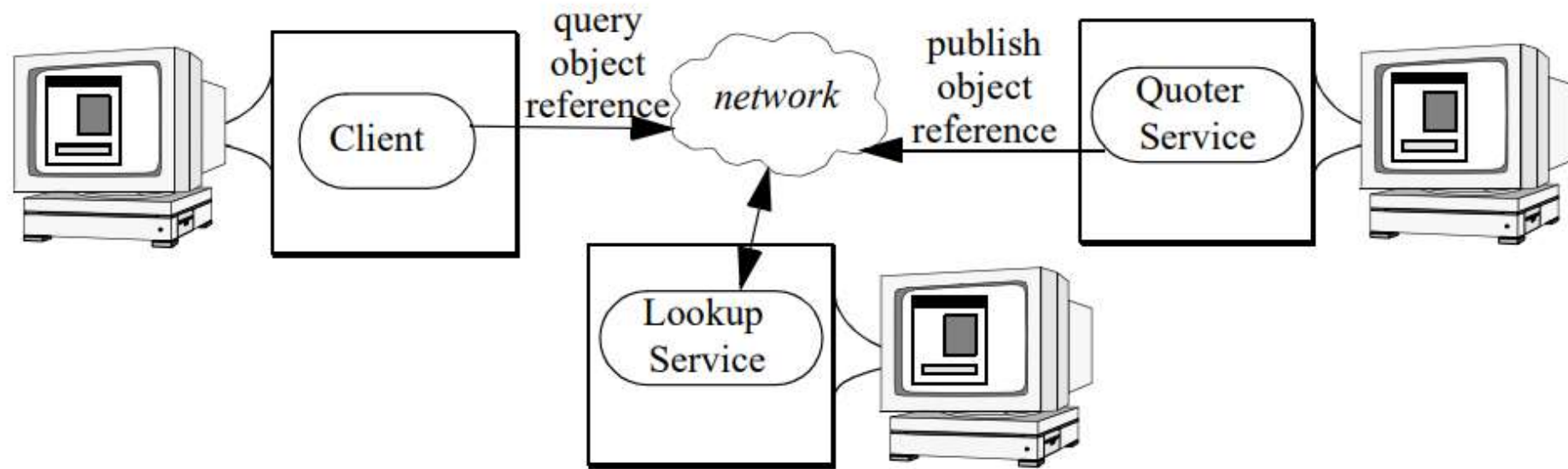
Někdo pronajímá něco někomu na nějakou dobu.



CORBA

- Standard pro usnadnění komunikace mezi systémy z různých platforem
- Systémy lišící se operačními systémy, prog. jazyky, hw
- Používá objektově orientovaný model, ale samotné systémy nemusí být objektově orientované

Příklad: CORBA



„Systems where resource usage needs to be controlled to allow timely release of unused resources.”

Implementace

1. Urči zdroje příslušící pronájmu

Implementace

1. Urči zdroje příslušící pronájmu
2. Urči policie pro vytvoření pronájmu

Implementace

1. Urči zdroje příslušící pronájmu
2. Urči policie pro vytvoření pronájmu
3. Urči povinnosti pronájmu

Implementace

1. Urči zdroje příslušící pronájmu
2. Urči policies pro vytvoření pronájmu
3. Urči povinnosti pronájmu
4. Urči policies pro expiraci pronájmu

```

class LookupService_impl : public POA_Lookup::LookupService {
public:
    Lease_ptr bind(const CosNaming::Name& name, CORBA::Object_ptr object, const TimeValue& time) {
        if (this->is_time_acceptable(time)) {
            // Create a new lease
            Lease_Impl* lease_impl = new Lease_Impl(time, this->_this(), name);
            // Get the CORBA object reference
            lease = lease_impl->_this();
            // Add to cache
            _cache.insert(lease);
            return lease;
        }
        return object;
    }

    void unbind(const CosNaming::Name& name) {
        // Delegate to the NS
        nameService_>unbind(name);
        clean_up();
    }

    TimeValue& renegotiate(LookupService::Lease_ptr lease, const TimeValue& time) {
        // Renegotiate lease
    }

    // ...

```

Kód ne jako obrázky, ale jako text !

jen zajímavé fragment, ne omáčku


```
class Lease_Impl : public POA_Lookup::Lease, Reactor::EventHandler {
public:
    void renew(const TimeValue& time) {
        // Renegotiate the lease with the grantor
        try {
            time_ = lookupService_->renegotiate(this->_this(), time);
            // Reschedule timer with the reactor for the duration of the lease
            reactor_->unregister_timer(this);
            reactor_->register_timer(time, this);
        }
        catch (...) {
            // ...
        }
    }

    // Callback method
    void on_timer_expire() {
        lookupService_->unbind(name_);
        valid_lease_ = false;
    }

    // Called by the lease holder to cancel a lease.
    void cancel() {
        // Cancel timer with the reactor
        reactor_->unregister_timer(this);
        this->on_timer_expire();
    }
    // ...
}
```



- Jednoduchá správa zdrojů
- Efektivní využití zdrojů
- Snadný update zdrojů
- Zlepšení spolehlivosti systému



- Dodatečný overhead
- Dodatečná aplikační logika
- Nutnost timeru

Leasing variants

- Auto renewal - by lease or holder or separate object
- No expiration - forever, holder cancels later
- Callbacks - callback on holder to inform about expiring lease
- Explicit Invalidation - Grantor sends signal about invalid object to holders. Holders then close lease.

Zdroje

- <https://www.ksi.mff.cuni.cz/~svoboda/courses/182-NSWI090/> respektive <http://www.earchiv.cz/l226/>
- <http://kircher-schwanninger.de/michael/publications/Leasing.pdf>
- https://hillside.net/plop/plop2001/accepted_submissions/PLoP2001/pjain1/PLoP2001_pjain1_1.pdf