



# LIETOJUMPROGRAMMU SASKARŅU IZSTRĀDE AR GraphQL

Autors: Andrejs Jurčenoks | aj05044 | 2020  
Darba vadītājs: doc., Dr.dat. Maksims Kravcevs

**PROBLĒMA** Pieaugot datu nozīmei, datu apmaiņas intensitātei, datu izmantotāju prasībām, palielinās nepieciešamība pēc augstas veiktspējas rīkiem, kas šādu apmaiņu nodrošina, tajā skaitā programmiski, caur lietojumprogrammu saskarnēm (*application programming interfaces, API*). Pēdējos divu gadu desmitos valdošā API uzbūves arhitektūra ir *Representational State Transfer, REST*. Pēdējā piecgadē arvien lielāku popularitāti iegūst GraphQL tehnoloģija

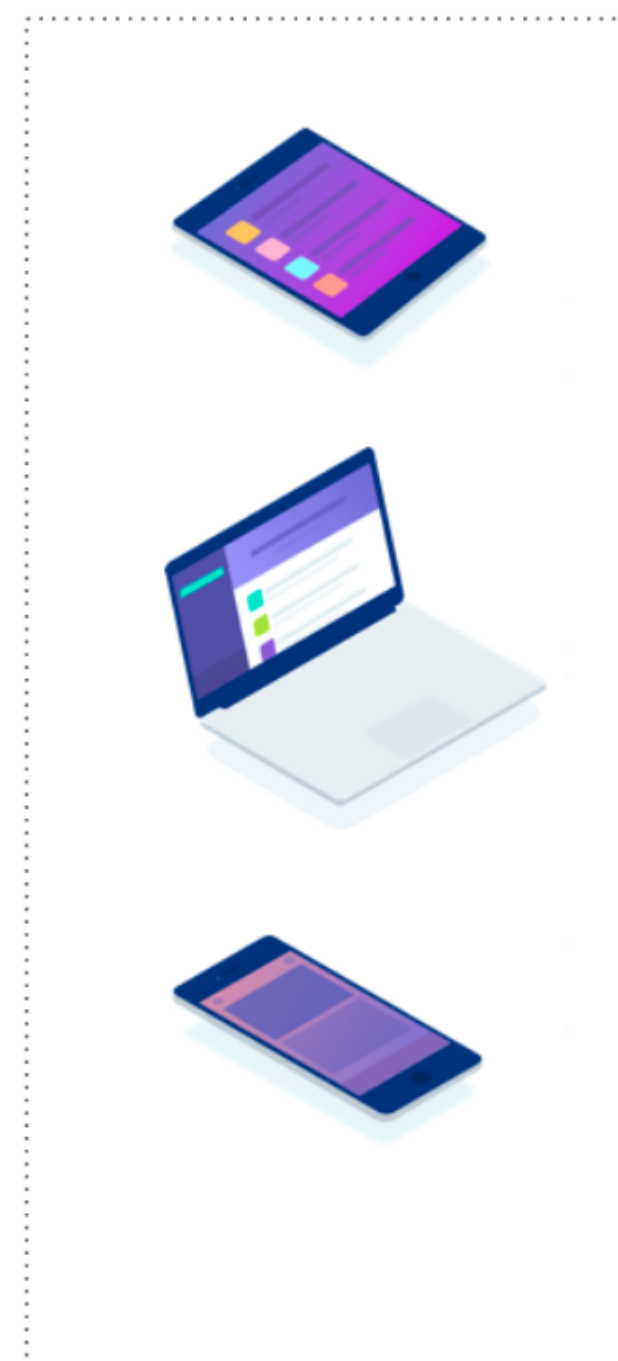
**MĒRĶIS** Izpētīt GraphQL, tā piedāvātas iespējas klienta pusē un nepieciešamās darbības servera pusē, vispārīgus priekšrocības un trūkumus, tajā skaitā tipiskās drošības un optimizācijas problēmas, saprast ieviešanas vai migrācijas pamatotību

**REZULTĀTI** Izpētītas vaicājumu un modifikāciju pieprasījumu sastādīšana klienta pusē un tiem nepieciešamas darbības servera pusē; tipiski optimizācijas problēmas (N+1 izsaukums) un pārslodzes ierobežošanas pasākumi, norādīti vispārīgas priekšrocības un trūkumi un ieteicami lietošanas scenāriji

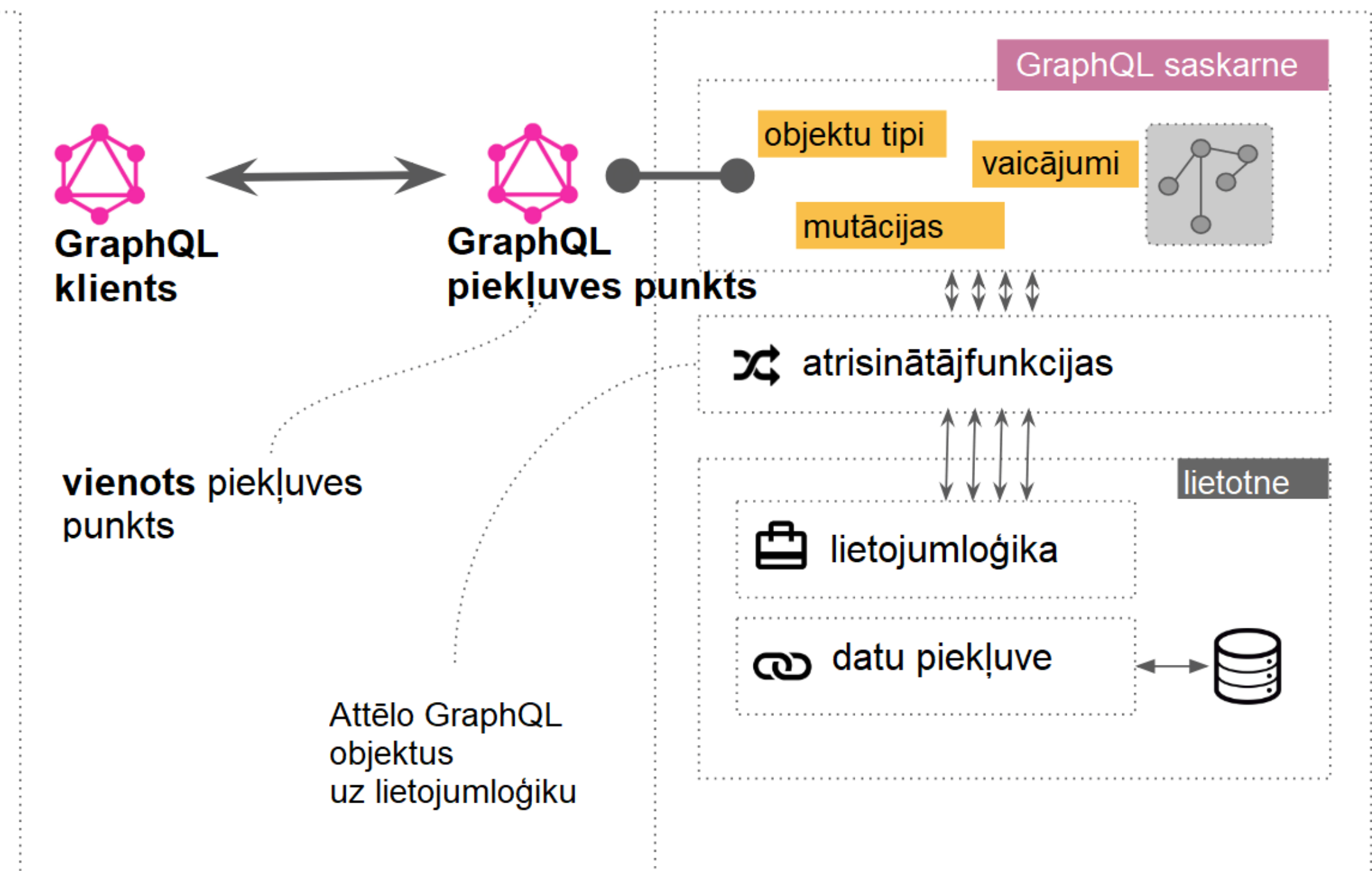
**SECINĀJUMI** labs risinājums scenārijos, kur REST neoptimāls; pie hierarhiskām vai grafa datu struktūrām; ērts vairāku datu avotu apvienošanai; ja jāsamazina tīkla izsaukumu skaits (jo iespējams iegūt visu nepieciešamo vienā izsaukumā) un datu pārraides apjomi (jo iespējams pieprasīt tikai nepieciešamo); ja API klienti ir dažāda tipa, un lielākais ieguvums mobilām ierīcēm; viens piekļuves punkts ar elastīgu versionēšanu - ērts piebiežām API kontrakta izmaiņām; ir jānodrošinās pret ārkārtīgi lieliem datu pieprasījumiem; jāoptimizē datu pieprasījumi (jānovērš N+1 problēma)

**TURPMĀKAIS DARBS** turpināt izpēti, izstrādājot kopīgam datu avotam dažādas arhitektūras piekļuves punktus: REST, OData un GraphQL tehnoloģijās, salīdzinot tehnoloģijas savā starpā un teorētiskā pētījuma rezultātus ar praktiskiem novērojumiem

## Priekšgalsistēma



## Aizmugursistēma

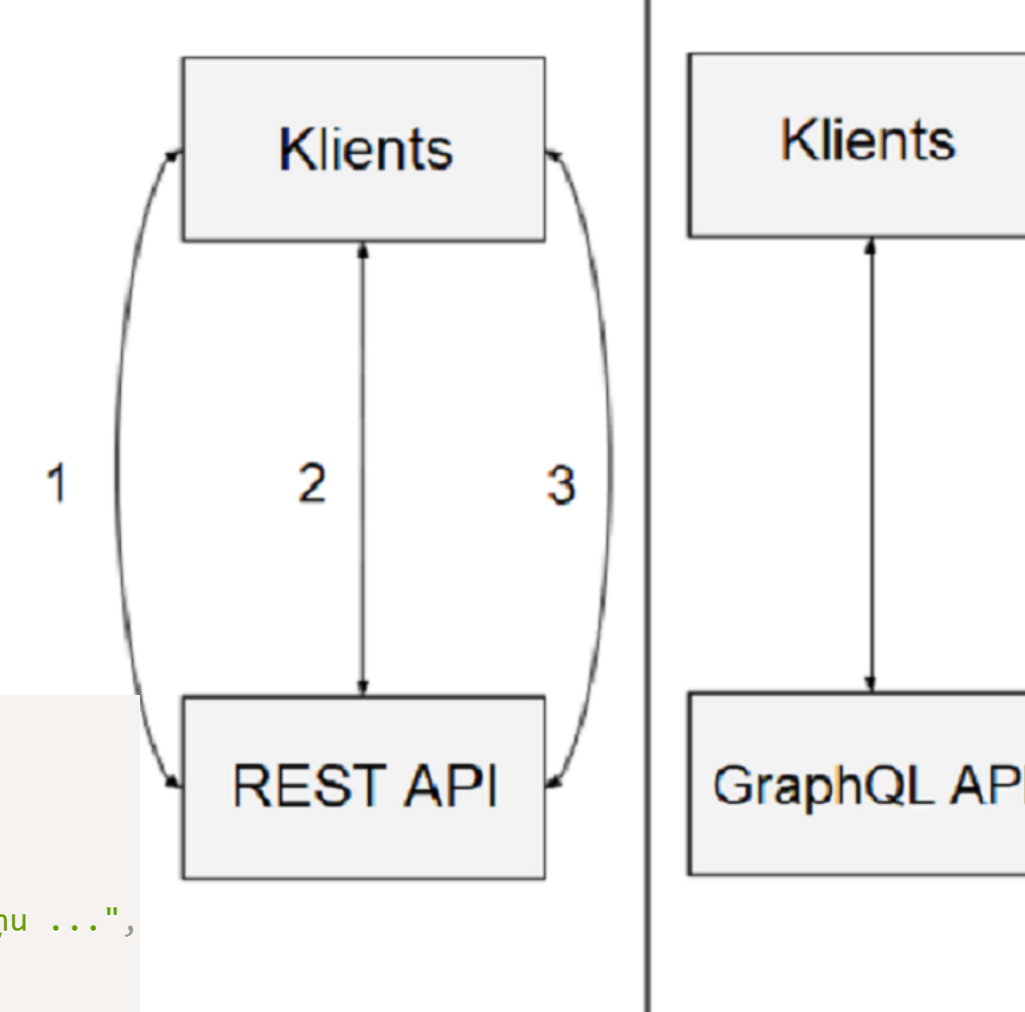


```
// 1. GET /magstudents/aj05044
{
  "Vārds": "Andrejs Jurčenoks",
  "KursaDarbs": "f20b8babb40a",
  "Adrese": "Rīga", // <-- nevajadzīgie dati
  "Apliecība": "aj05044", // <-- nevajadzīgie dati
  "StudijuKursi": [ // <-- nevajadzīgie dati
    "b517ab71582d", "5cc5b802bcc5", ...
  ]
}

// 2. GET /magcourseworks/f20b8babb40a
{
  "Id": "f20b8babb40a",
  "Nosaukums": "Lietojumprogrammu saskarņu izstrāde ar GraphQL",
  "Anotācija": "Kursa darbā tiek izpētīta lietojumprogrammu saskarņu ...",
  "Vadītājs": "1a9afaf4c169"
}

// 3. GET /staff/ba9afaf4c169
{
  "Vārds": "Maksims Kravcevs",
  "Amats": "doc.",
  "ZinatniskaisGrāds": "Dr.dat.",
  "Adrese": "Rīga" // <-- nevajadzīgie dati
}
```

// REST alternatīva - atsevišķs piekļuves punkts:  
GET /magcourseworksByAuthor/aj05044  
// trūkumi serveram: savu adresi katram vaicājumam?  
// trūkumi klientam: iespējami nevajadzīgi dati, gan trūkstoši dati



```
// vienīgais vaicājums
query {
  Vārds
  KursaDarbs {
    Nosaukums
    Vadītājs {
      Vārds
      Amats
      ZinatniskaisGrāds
    }
  }
}

// atbilde:
{
  "Vārds": "Andrejs Jurčenoks",
  "KursaDarbs": {
    "Nosaukums": "Lietojumprogrammu saskarņu izstrāde ar GraphQL",
    "Vadītājs": {
      "Vārds": "Maksims Kravcevs",
      "Amats": "doc.",
      "ZinatniskaisGrāds": "Dr.dat."
    }
  }
}
```