



KADA (Drupal 7) migraatio uuteen (versioon) webiin

Hallittu elinkaaren siirto suoran migraation sijaan

Mikko Malmgren & Antti Tuppurainen

Mikko Malmgren / Kuntaliitto
@mikko_malmgren

Antti Tuppurainen / Industry62
@wroxbox

23.4.2019

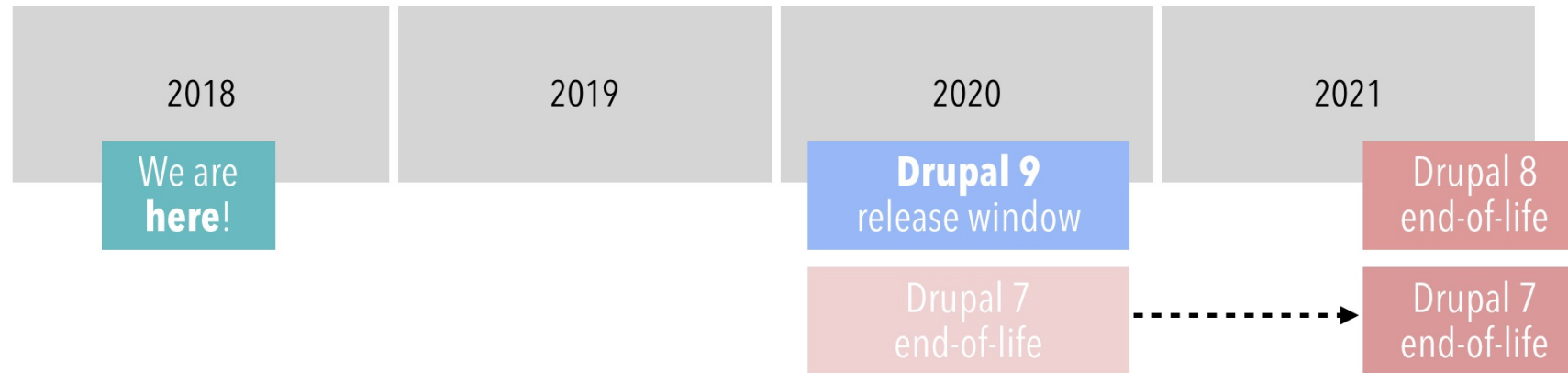
Selvityksen tausta ja tarve

- Useat kunnat ovat hyödyntäneet Turun kaupungin Drupal 7 verkkopalveluiden alustaa ja toteutusta (KADA) omien palveluiden perustana.
- Drupal 7 edustaa vanhaa versiota ja tulevaisuudessa on edessä siirtyminen uuteen Drupal 8/9 versioon tai ainakin uudistaa verkkopalvelun visuaalinen ilmettä tai joitakin toiminnallisia osia.
- Suora siirtyminen Drupal 8/9 versioon vaatii merkittävää panostusta, koska uudet versiot ovat teknisesti täysin uudistettuja ja taaksepäinyhteensopivuus on katkaistu.
- KADAssa on contrib-moduuleita kymmeniä sekä paketoituja featureita. Kaikki sellaisenaan vaikeasti, jos ollenkaan, hyödynnettävissä uudessa versiossa.
- Featurepohjainen paketointi on myös Drupal 7– ominaista ja taas Drupal 8/9 on oma CMI –pohjainen yhtenevä konfiguraationhallinta.

Drupal 7 – End Of Life

Drupal 7

will be supported until **November 2021**



<https://dri.es/drupal-7-8-and-9>

- Kada – koodi on avoimesti saatavilla Githubissa. Koodia on hyödynnetty (forkattu), mutta uusia toiminnallisuuksia ei ole palautettu (kontribuoitu) takaisin perusversioon.
- Näkyvyyttä toteutusten yhdenmukaisuuteen ei ole. Jokainen toteutus voi poiketa merkittävästi alkuperäisestä Kadasta.
- Myöskään Turun verkkopalvelut eivät välttämättä vastaa enää Kadaa.
- Tiedonhallintamalli saattaa poiketa installaatioissa
- Kukin verkkopalvelu on toiminnallisuudeltaan erilainen (teema)

Suoraa yhteistä ohjeistusta suoraan migraatioon on hankala määritellä, johtuen mm edellä mainituista syistä. Tästä syystä ehdotamme seuraavaa...

Suoran migraation sijaan hallittu siirtyminen ja kehittäminen uuteen.

Nykyisen ratkaisun elinkaarta voidaan pidentää ja siirtyminen uuteen versioon on mahdollista toteuttaa pidemmän ajan kuluessa ja hallitusti seuraavan ehdotuksen mukaisesti:

- Rajataan nykyinen Drupal 7 installaatio erillisen palomuurin taakse ja tarjotaan data rajapinnan kautta toisiin järjestelmiin
 - Huolehditaan vain security –päivityksistä.
 - Ylläpitäjillä säilyy vanha tuttu järjestelmä ja migraatiot muihin järjestelmiin säilyvät kun kaikkea ei tarvitsekaan kirjoittaa uusiksi.
- Data exportoidaan toisten järjestelmien käyttöön esimerkiksi seuraavilla tavoilla:
 - Custom JSON output moduuli
 - Views Datasource as REST JSON
 - Services

*Ajatellaan nykyistä KADA-pohjaista järjestelmää
API:nä muiden joukossa. Tämä ajattelu vapauttaa
yhden monoliittisen järjestelmän kehittämisestä
monipuolisemmaksi ja arkkitehtuuriltaan
vapaammaksi.*

Mitä ovat tämän päivän kunnan verkkopalvelut?

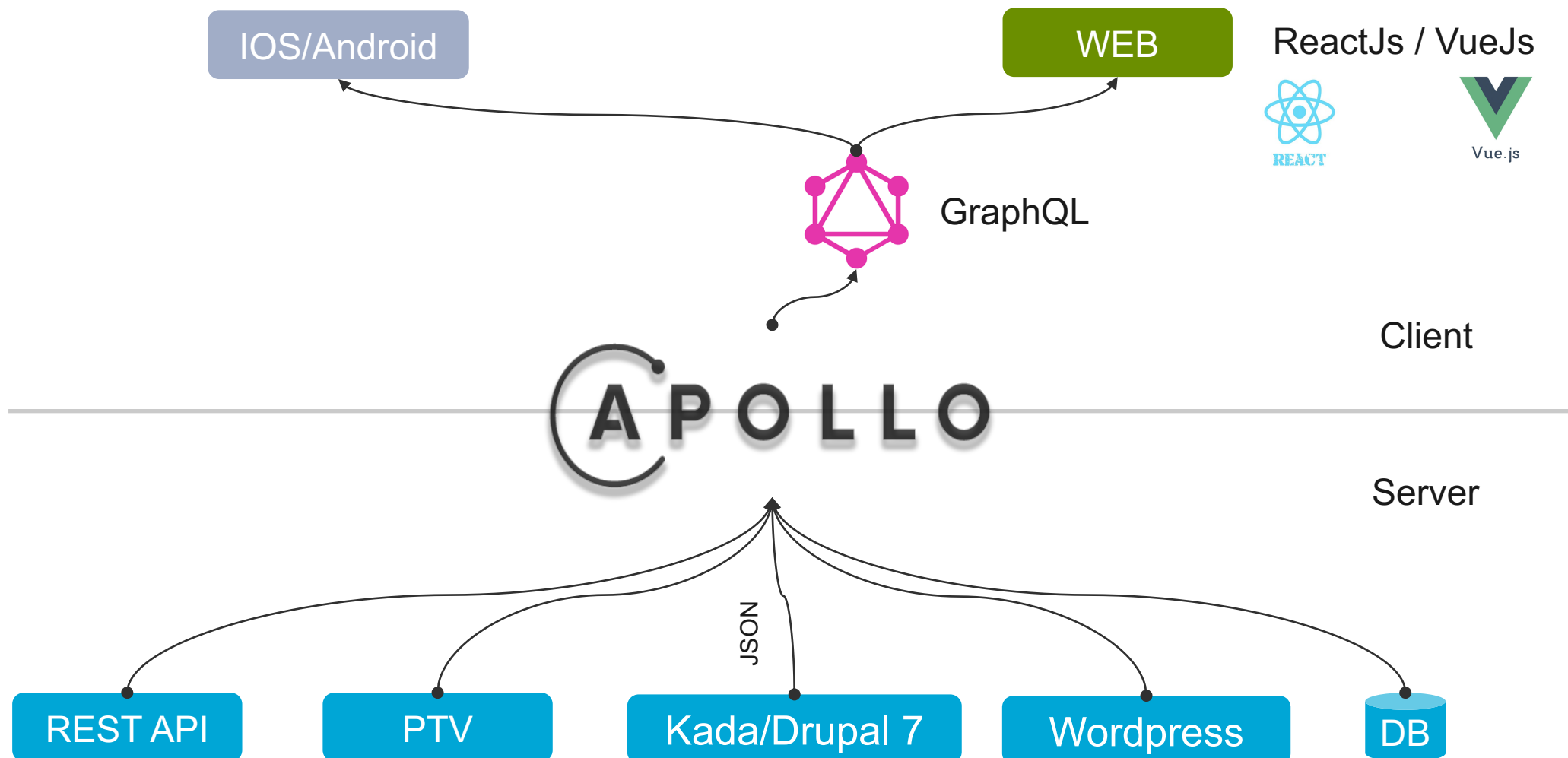
- Uusia tapoja rakentaa verkkopalveluja on olemassa muitakin kuin Wordpress ja Drupal. Kaiken ei tarvitse olla yhdessä monoliittisessä sisällönhallintajärjestelmässä.
- Sähköiset asiointikanavat joita aikoinaan tahdottiin kaikki kunnan verkkopalveluun onkin kehitetty kutakin omaan käyttötarkoitukseen ja näin ollen usein verkkosivuilta onkin vain linkki asiointikanaviin ja verkkosivut ovatkin muodostuneet viestintäkanavaksi.
- Tämä pitää ottaa huomioon ja siksi onkin helpompi esitellä seuraava tapa eri tietolähteiden hyödyntämiseen.

Headless, GraphQL ja Drupal -installaatiot

- Hyödynnetään GraphQL teknologiaa, joka mahdollistaa tietosisältöjen hyödyntämisen eri lähteistä.
- Luodaan perusta uudella teknologialla (esim. Kuntafibase) ja ja yhdistetään uusi alusta vanhaan GraphQL:n avulla ja aloitetaan elinkaaren siirto / palvelun kehittäminen uudessa ympäristössä uusilla menetelmillä.
- Uudet verkkosivut voivat täten olla esimerkiksi seuraavanlaisia hybridejä:
 - APIT > Apollo server > Vue/React frontend
 - APIT > Migrate API/Drupal 8 > Drupal 8
 - APIT > Migrate API/Drupal 8 > Tome
 - APIT > Migrate API/Drupal 8 > Gatsby
 - APIT + Drupal 7 + Drupal 8 > Apollo Server > Vue/React frontend

(Migrate API/Drupal 8 == KuntafiBase)

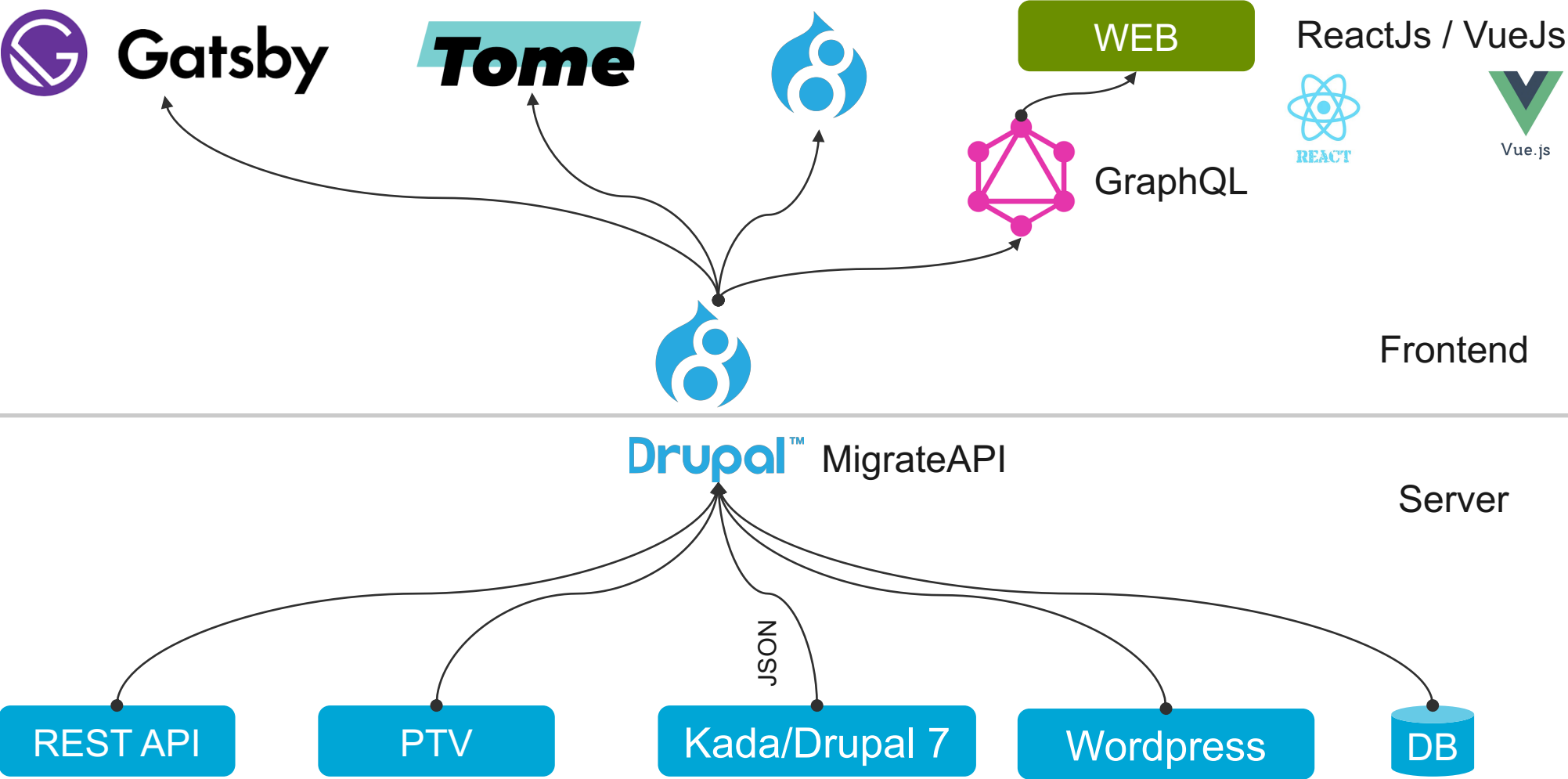
Esimerkki 1: Apollo Server



<https://www.apollographql.com/>

<https://medium.com/@dwalsh.sdlr/gatsby-apollo-graphcool-netlify-the-webs-promised-land-6dd510efbd72>

Esimerkki 2: KuntafiBase

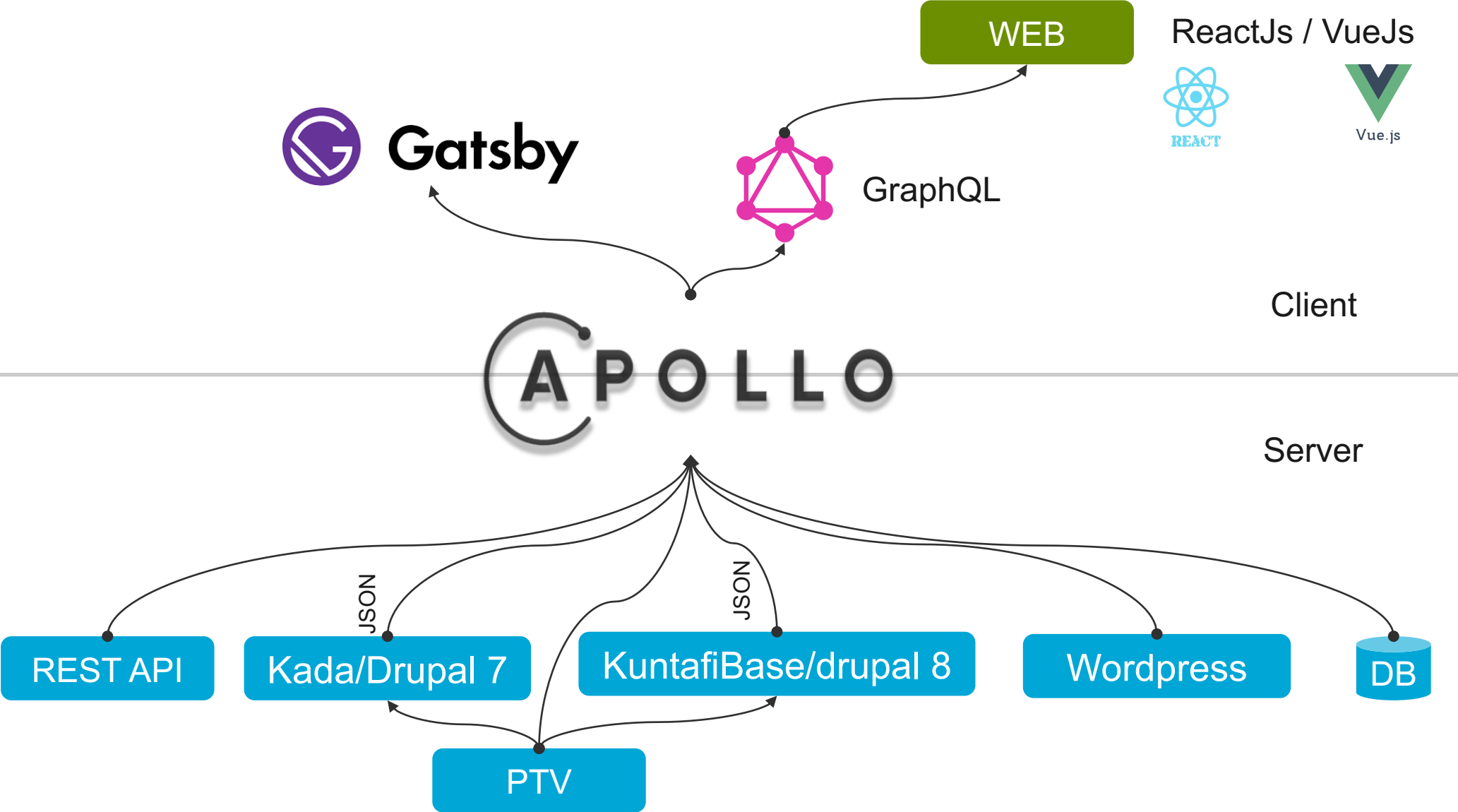


<https://www.gatsbyjs.org/>

<https://tome.fyi/>

<https://www.slideshare.net/Gatsbyjs/beyond-static-building-dynamic-apps-with-gatsby>

Esimerkki 3: KuntafiBase



- Esimerkit näyttävät miten voimme hyödyntää vanhoja järjestelmiä uusien järjestelmien/metodien datalähteenä. Datat voidaan hakea suoraan GraphQL –plugineilla Drupaliin joka renderöi ne verkkopalvelun käyttäjille. Tämä on suoraviivaisin tapa. Usein suoraviivaisin ei kuitenkaan tuota parasta lopputulosta ja siksi esitinkin vaihtoehtoisia lopputuloksia perinteisen Drupal/Wordpress –teeman ohelle.
- Hakukoneet suosivat nopeita verkkosivuja. Käyttäjät olettavat saavansa nopeat ja helposti toimivat verkkosivut. Hakumoottoreiden oletetaan löytävän täsmällisesti tarvittava sisältö. Nämä kaikki oletukset on mahdollista saavuttaa edellä kuvatuilla staattisen generoinnin työkaluilla tai vue/react –tyyppisillä frontendeilla.