cloud connect

A cloud native company

Latihan Aplikasi Moden

Jabatan Digital Negara

9 - 12 September 2024

cloud connect

Material Latihan:
https://bit.ly/jdn-lpam

Google Meet:
https://bit.ly/jdn-lpam-meet1

Latihan Hari Kedua

cloud connect

HARI KEDUA	
Sesi	Topik
Sesi 1	Ringkasan Latihan Hari Pertama
Sesi 2	Fastify Validation & Serialization
Sesi 3	Pengenalan kepada PrismaORM
Sesi 4	Pembangunan API untuk MySQL Database sediada
Sesi 5	Pembangunan API untuk MySQL Database baru
Sesi 6	Pengenalan kepada Jamstack Framework
Sesi 7	Pengenalan kepada Vue.js

Ringkasan Latihan Hari Pertama

cloud connect

Hari Pertama	
Sesi	Topik
Sesi 1	Pengenalan Latihan
Sesi 2	Pengenalan kepada teknologi aplikasi moden & Jamstack Framework
Sesi 3	Pengenalan kepada Microservices & API
Sesi 4	Pengenalan kepada Fastify
Sesi 5	Fastify Server, Logging, Environment Variables, Errors, Routes

Fastify Routes

cloud connect

- The route methods will configure the endpoints of your application.
- Two ways to declare a route with Fastify:
  - Full declaration
    - `fastify.route(options)`
  - Shorthand declaration
    - `fastify.get(path, [options], handler)`

Fastify Routes - Full declaration

cloud connect

- `fastify.route(options)`
- Options:
  - method: currently it supports 'DELETE', 'GET', 'HEAD', 'PATCH', 'POST', 'PUT', 'OPTIONS', 'SEARCH', 'TRACE', 'PROPFIND', 'PROPPATCH', 'MKCOL', 'COPY', 'MOVE', 'LOCK', 'UNLOCK', 'REPORT' and 'MKCALENDAR'.
  - url: the path of the URL to match this route (alias: path).
  - schema: an object containing the schemas for the request and response. They need to be in JSON Schema format.
    - body: validates the body of the request if it is a POST, PUT, PATCH, TRACE, SEARCH, PROPFIND, PROPPATCH or LOCK method.
    - querystring or query: validates the querystring. This can be a complete JSON Schema object, with the property type of object and properties object of parameters, or simply the values of what would be contained in the properties object as shown below.
    - params: validates the params.
    - response: filter and generate a schema for the response, setting a schema allows us to have 10-20% more throughput.

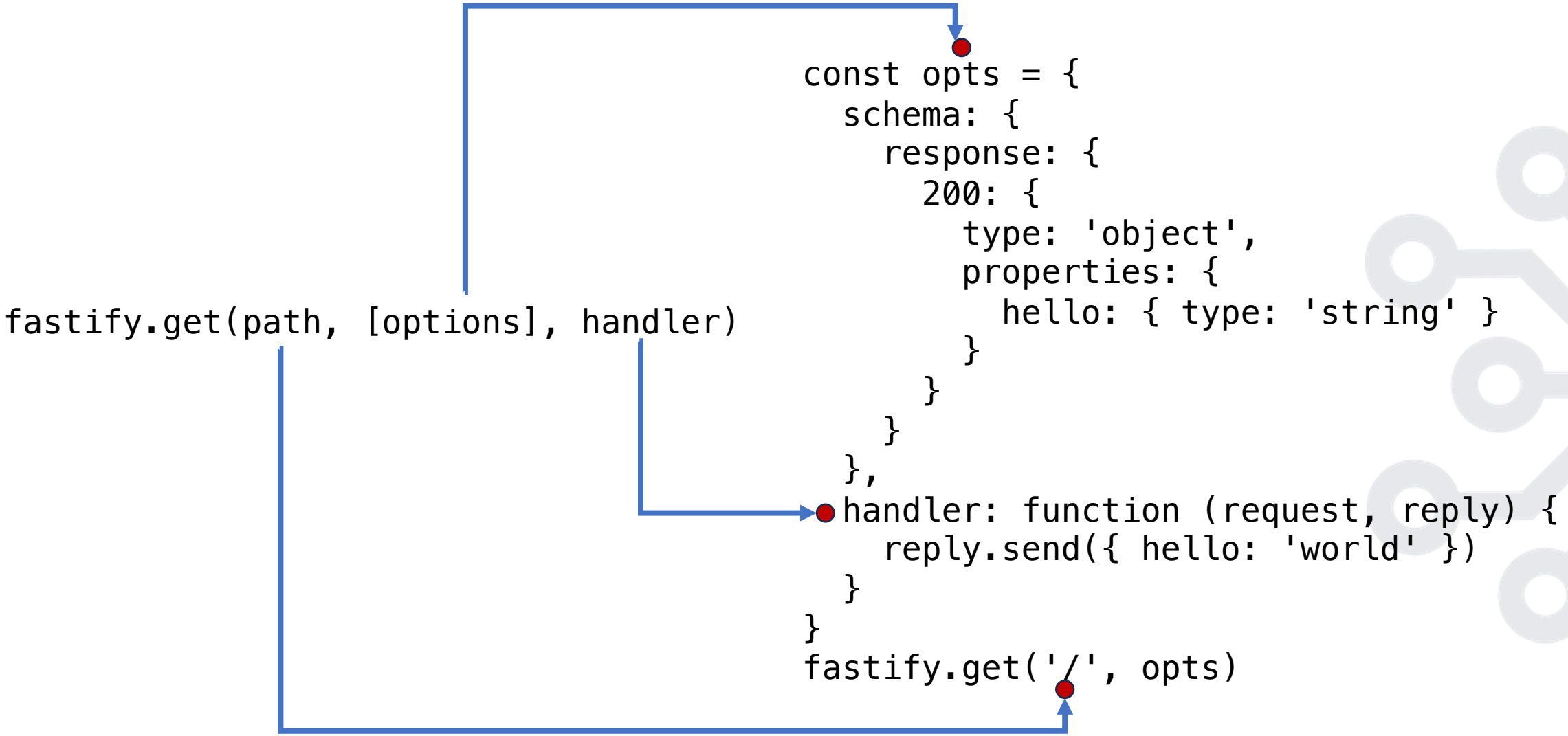
Fastify Routes - Shorthand declaration

cloud connect

- `fastify.get(path, [options], handler)`
- `fastify.head(path, [options], handler)`
- `fastify.post(path, [options], handler)`
- `fastify.put(path, [options], handler)`
- `fastify.delete(path, [options], handler)`
- `fastify.options(path, [options], handler)`
- `fastify.patch(path, [options], handler)`

Fastify Routes - Shorthand declaration

cloud connect



Validation & Serialization

cloud connect

- Fastify uses a schema-based approach, and even if it is not mandatory we recommend using **JSON Schema** to validate your routes and serialize your outputs.
- Internally, Fastify compiles the schema into a highly performant function.
- Validation will only be attempted if the content type is application-json, as described in the documentation for the content type parser.
- The validation and the serialization tasks are processed by two different, and customizable, actors:
  - **Ajv v8** for the validation of a request
  - **fast-json-stringify** for the serialization of a response's body

Validation

cloud connect

- The route validation internally relies upon Ajv v8 which is a high-performance JSON Schema validator.
- The supported validations are:
  - **body**: validates the body of the request if it is a POST, PUT, or PATCH method.
  - **querystring** or **query**: validates the query string.
  - **params**: validates the route params.
  - **headers**: validates the request headers.
- All the validations can be a complete JSON Schema object (with a type property of 'object' and a 'properties' object containing parameters) or a simpler variation in which the type and properties attributes are forgone and the parameters are listed at the top level

Serialization

cloud connect

- Send your data to the clients as JSON, and Fastify using **fast-json-stringify**, to provide an output schema in the route options.
- To use an output schema, as it can drastically increase throughput and help prevent accidental disclosure of sensitive information.

cloud connect

Latihan 6 –Fastify Validation & Serialization

cloud connect

Modern ORM with Prisma

Ringkasan *MySQL Database*

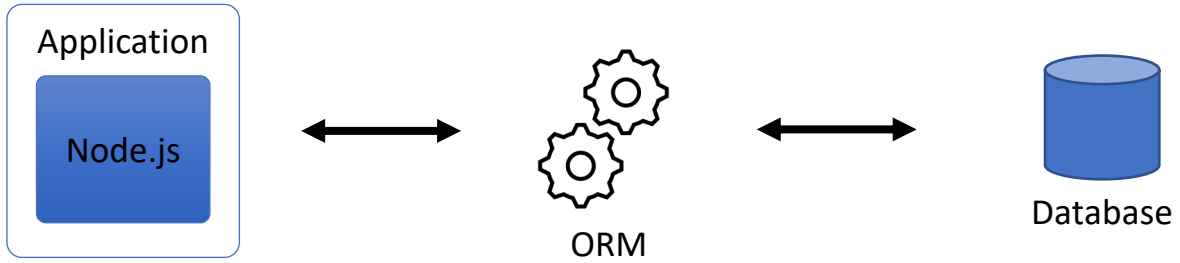
cloud connect

- MySQL is an open source, relational database management system (RDBMS) based on structured query language (SQL).
- Created by a Swedish company called MySQL AB in 1995.
- It stores data in tables, columns and rows.
- Most popular RDBMS.

Apa itu ORM?

cloud connect

- **O**bject **R**elational **M**apping - is a technique that creates a **layer** between the language and the database **without** having to write SQL queries.




```
SQL:  SELECT * FROM users WHERE email = 'test@test.com';

                        ORM:
          var orm = require('generic-orm-libary');
var user = orm("users").where({ email: 'test@test.com' });
```

Kelebihan ORM

cloud connect

- You get to write in the language you are already using.
- You don't have to be an expert in SQL queries, since you don't have to write SQL. Also, a lot of stuff is done automatically, hence Your code becomes short, clean, and easy to maintain
- It abstracts away the database system so that switching from MySQL to PostgreSQL, or whatever flavor you prefer.
- Depending on the ORM you get a lot of advanced features out of the box, such as support for transactions, connection pooling, migrations, seeds, streams, and all sorts of other goodies.
- Can potentially switch databases without any issues, as vendor-specific logic is abstracted in an ORM

NodeJS ORM

cloud connect

- Sequelize
  - supports PostgreSQL, MySQL, MariaDB, SQLite, and MSSQL.
- Knex.js
  - Support PostgreSQL, MSSQL, MySQL, MariaDB, SQLite3, Oracle, and Amazon Redshift
- Bookshelf.js
  - PostgreSQL, MySQL, and SQLite3
- Prisma (next-generation ORM)
  - PostgreSQL, MySQL, and SQLite3
  - MSSQL & MongoDB (preview)

- Eloquent - Laravel ORM
  - Support PostgreSQL, MySQL, SQLite, and MSSQL

Prisma ORM

cloud connect

- an open source next-generation ORM toolset for Node.js and TypeScript.
- provides a type-safe API for submitting database queries, returning plain old JavaScript objects
- to build Server-side applications typically are API servers that expose data operations via technologies like REST, GraphQL or gRPC.

Prisma

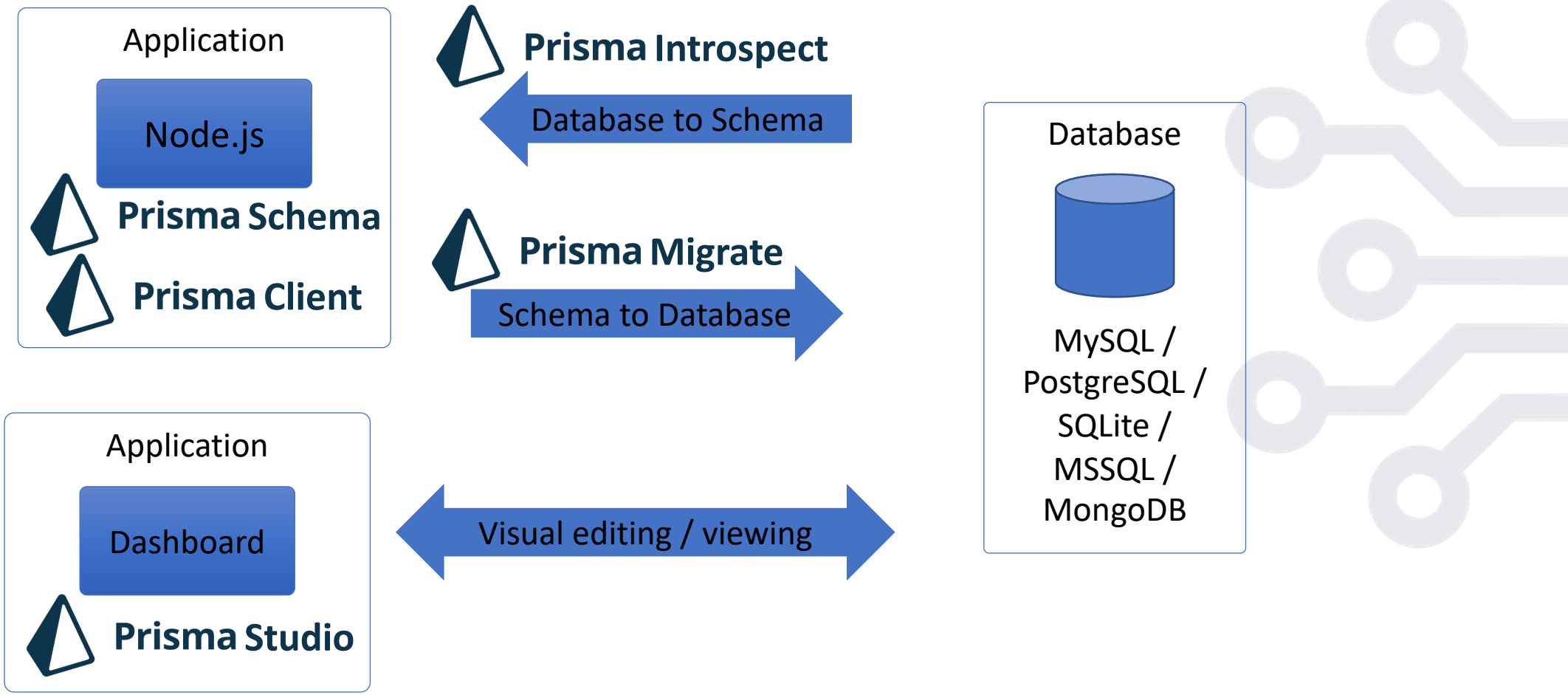
Komponen Prisma

cloud connect

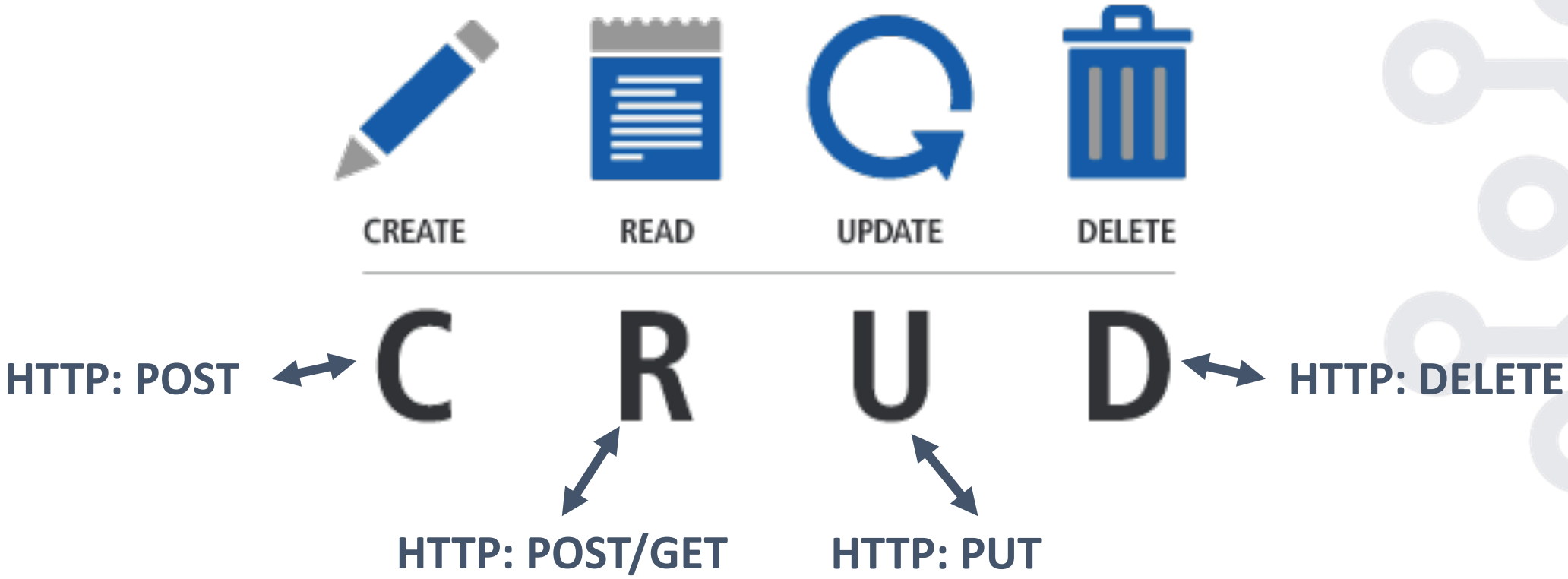
Prisma

- querying (with **Prisma Client**)
- data modeling (in the **Prisma schema**)
- auto-generate schema (with **Prisma Introspect**)
- migrations (with **Prisma Migrate**)
- prototyping (via prisma db push)
- seeding (via prisma db seed)
- visual viewing and editing (with **Prisma Studio**)

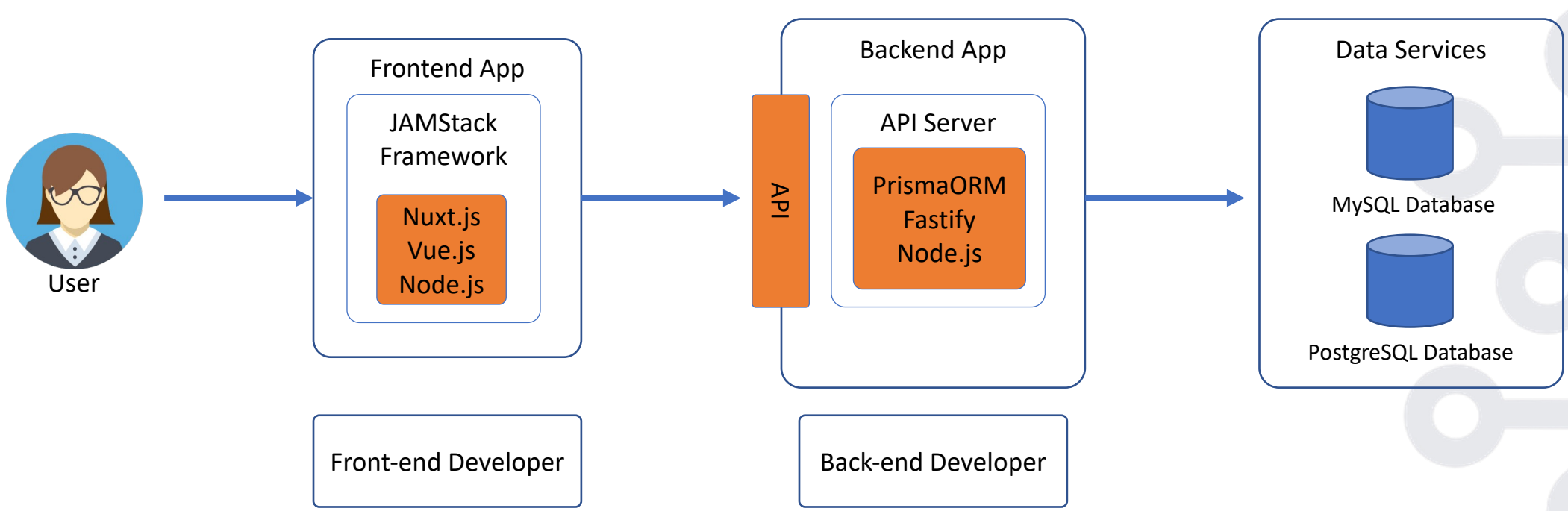
Komponen Prisma



How to expose an API?



Skop Latihan



Latihan 7 –Pembangunan API dari MySQL Database sediada

- Menggunakan MySQL dihoskan di *Public Cloud*
- Nama *database: testdb*

cloud connect

Latihan 8 –Pembangunan API dari MySQL Database yang baru

- Menggunakan MySQL dihoskan di Public Cloud
- Setiap peserta diberikan database: user1, user2….. userN

https://jamstack.org/

Jamstack

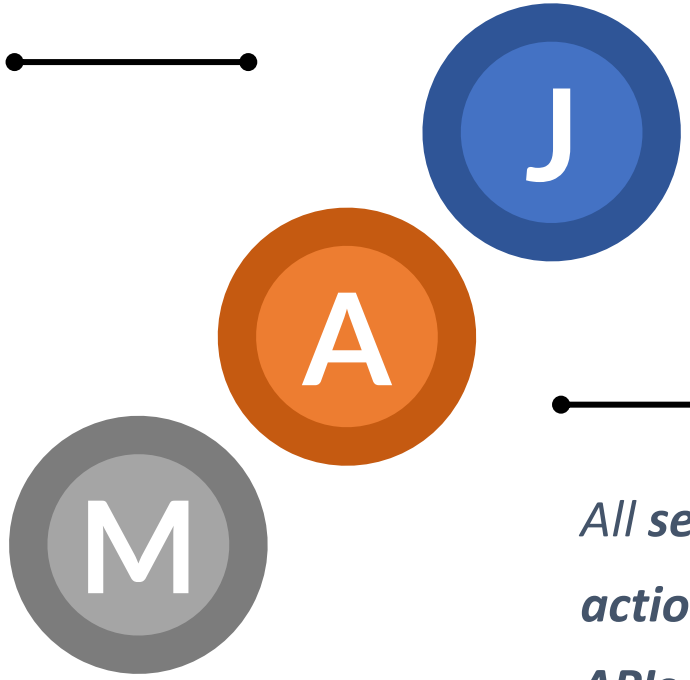
cloud connect

Javascript

*Any dynamic programming during the request/response cycle is handled by **JavaScript**, running entirely on the client.*

Markup

*Templated **markup** should be **prebuilt** at **deploy time**, usually using a site generator for content sites, or a build tool for web apps.*

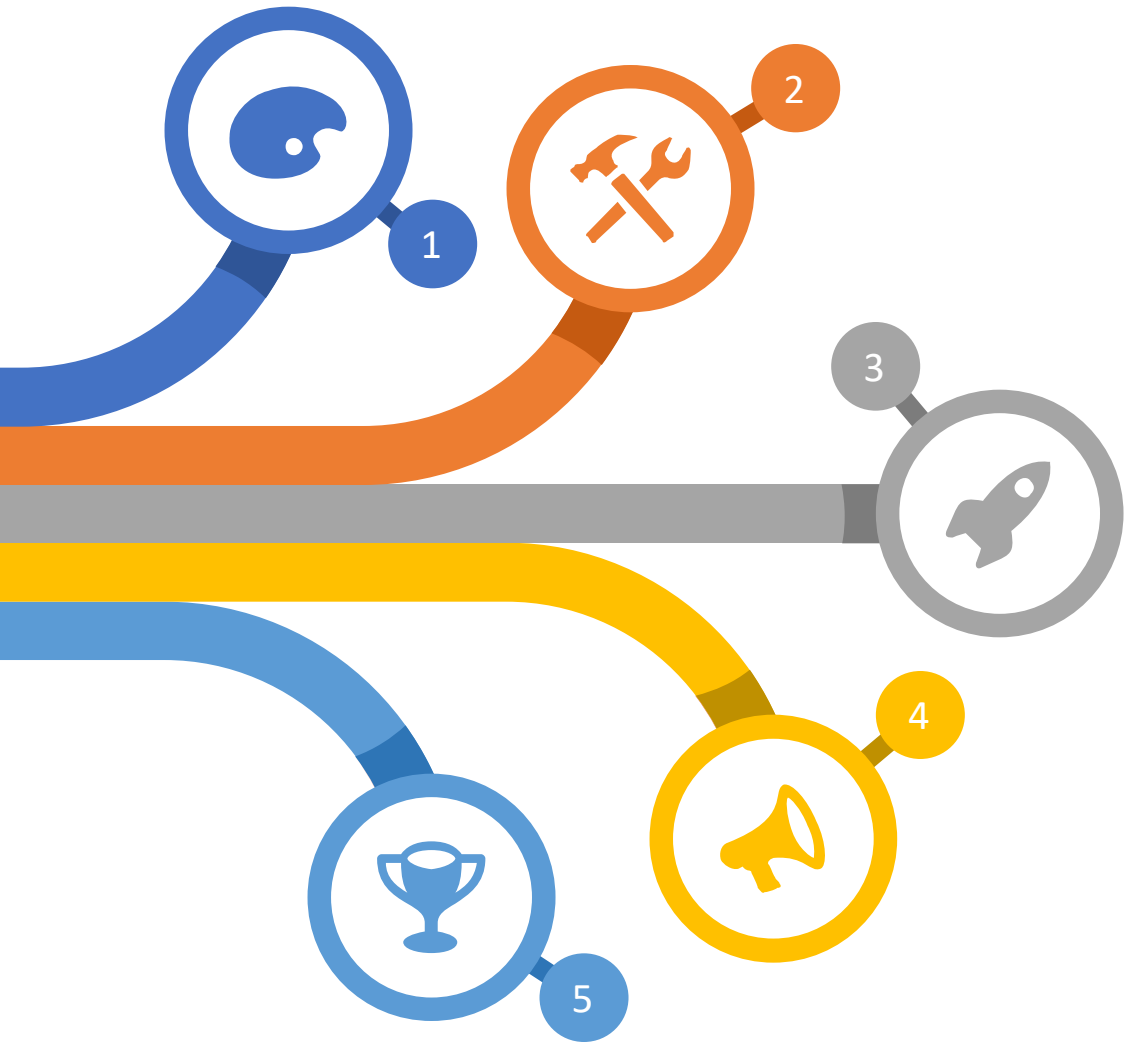


APIs

*All **server-side processes** or **database actions** are abstracted into reusable **APIs**, accessed over HTTPS with JavaScript.*

Kenapa *Jamstack*?

cloud connect



1

Faster and Better Performance

generate new pages at deploy time and serve pre-built markup and assets over a CDN.

2

Less expensive and easier to scale

lesser complexity of development reduces costs and also hosting of static files is cheap or even free

3

Higher Security

Static websites have a very low potential for vulnerabilities because it is just HTML files and external API-handling served over a CDN

4

Better developer experience

not difficult to learn. With just HTML, CSS and JS experience, developers can build complex websites

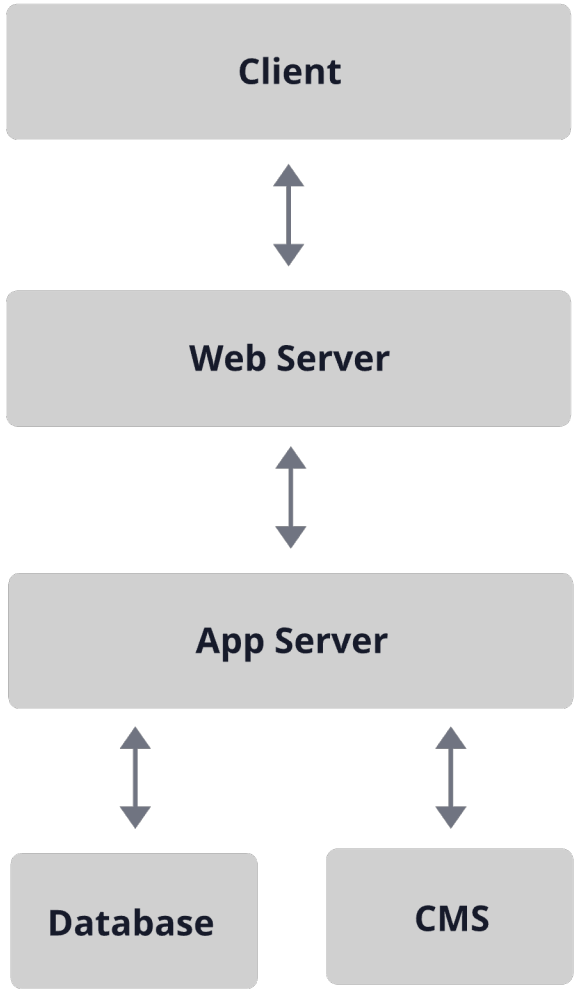
5

Great community

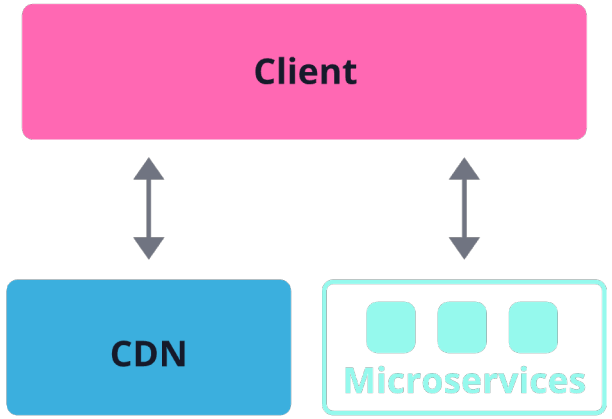
The JAMstack community is growing over time

Jamstack

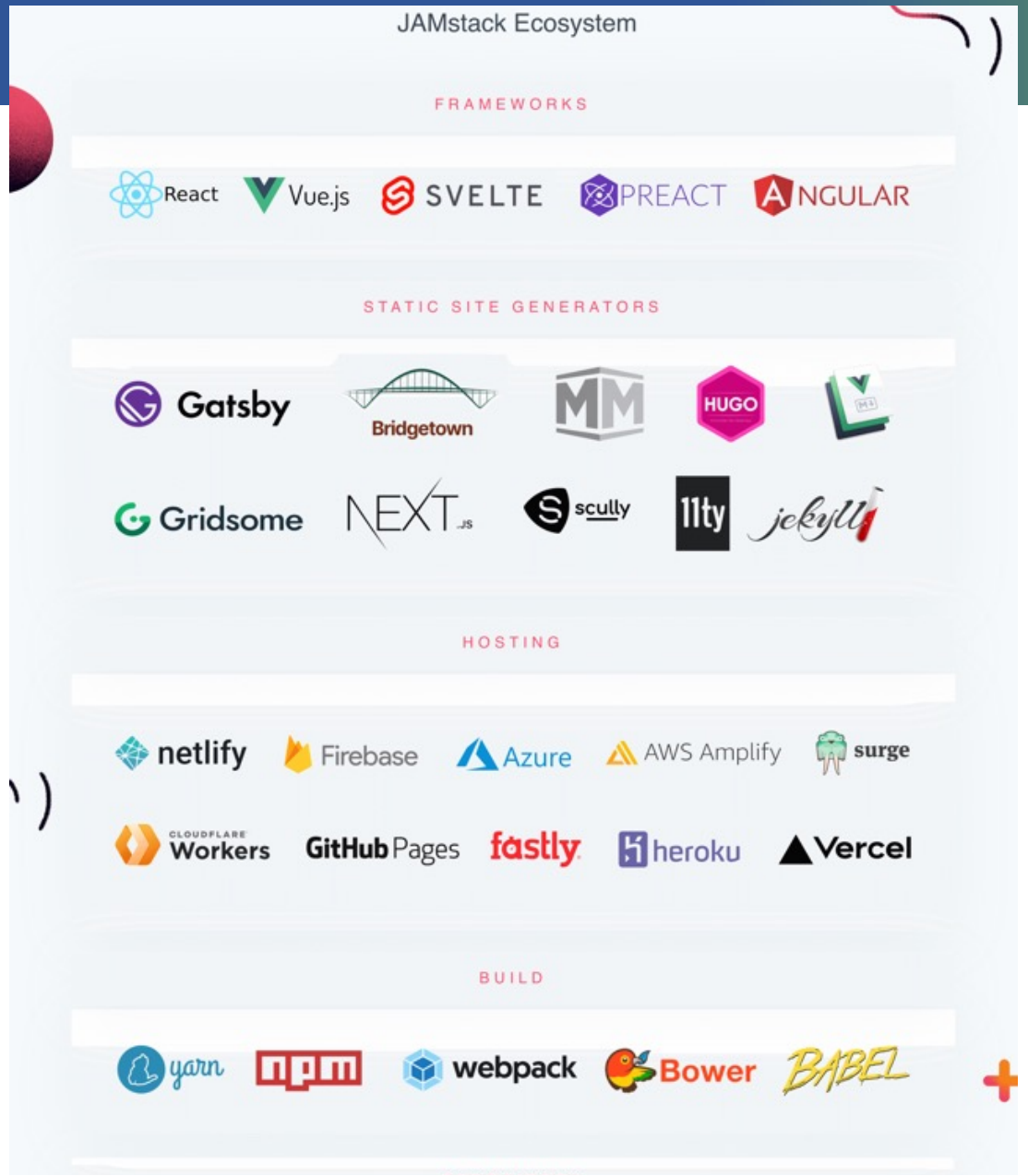
cloud connect



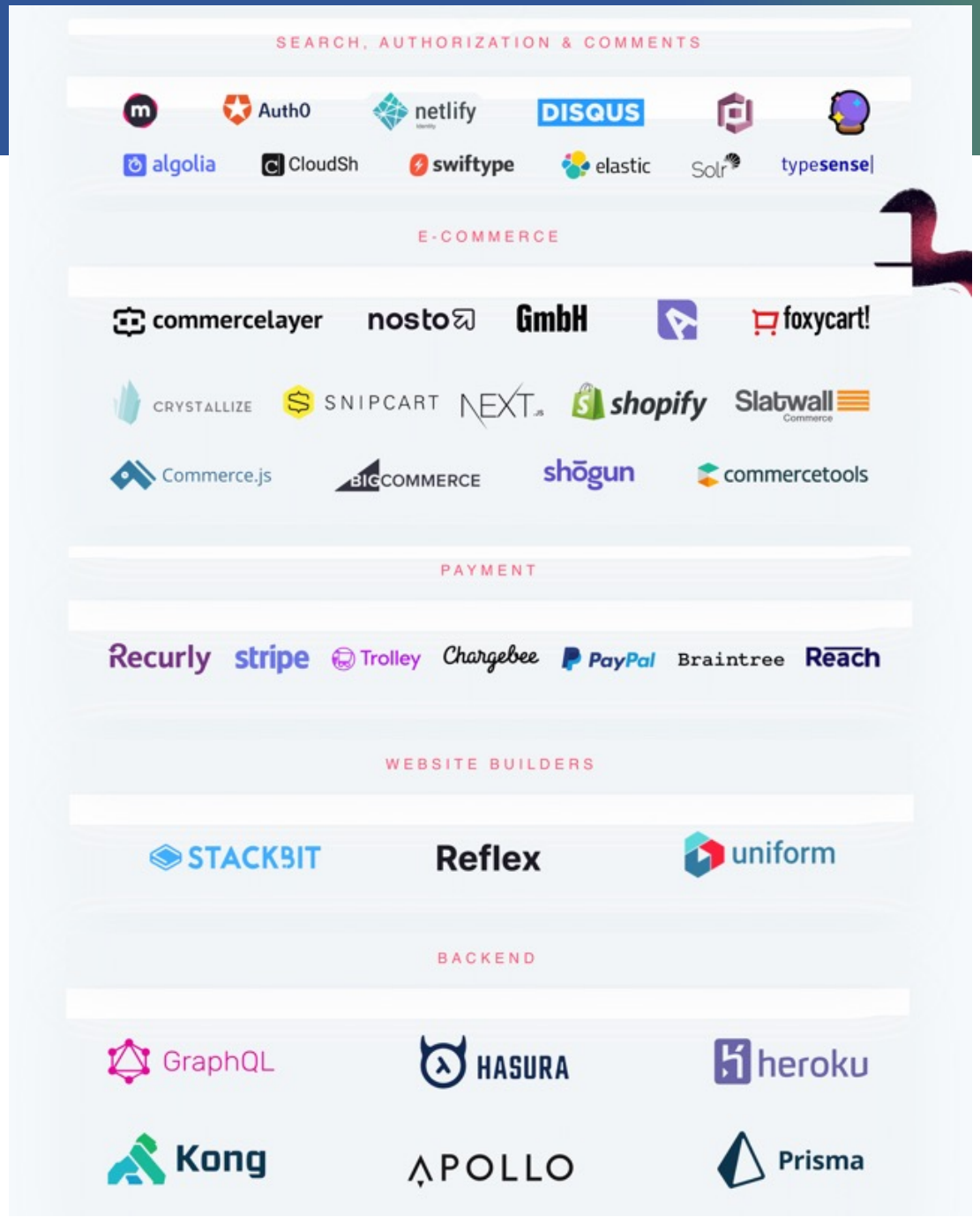
Traditional Web Apps



Jamstack


JAMstack Ecosystem
FRAMEWORKS
React
Vue.js
SVELTE
PREACT
ANGULAR
STATIC SITE GENERATORS
Gatsby
Bridgetown
MM
HUGO
Gridsome
NEXT.js
scully
11ty
jekyll
HOSTING
netlify
Firebase
Azure
AWS Amplify
surge
CLOUDFLARE Workers
GitHub Pages
fastly
heroku
Vercel
BUILD
yarn
npm
webpack
Bower
BABEL


REPOSITORY
git
GitHub
GitLab
Bitbucket
CONTENT MANAGEMENT
GIT-based
FORESTRY
CRAFTERCMS
netlifycms
Publii
FrontAid CMS
API-FIRST
SANITY
DATO CMS
strapi
TakeShape
contentful
storyblok
agility
graphcms
ButterCMS
Kontent.
prismic.io
magnolia
FUNCTIONS
netlify Functions
Azure Functions
Cloud Function for Firebase
Architect
UTILITY APIs
DATABASE
FAUNA
DynamoDB
8base
Sheetsu
mongoDB.
Airtable
FORMS
mailchimp
Paperform
Google Forms
netlify
FORMSPREE
Typeform
FormKeep
StaticForms


SEARCH, AUTHORIZATION & COMMENTS
Auth0
netlify
DISQUS
algolia
CloudSh
swiftype
elastic
Solr
typesense
E-COMMERCE
commercelayer
nosto
GmbH
foxycart!
CRYSTALLIZE
SNIPCART
NEXT.js
shopify
Slatwall
Commerce
Commerce.js
BIGCOMMERCE
shōgun
commercetools
PAYMENT
Recurly
stripe
Trolley
Chargebee
PayPal
Braintree
Reach
WEBSITE BUILDERS
STACKBIT
Reflex
uniform
BACKEND
GraphQL
HASURA
heroku
Kong
APOLLO
Prisma

https://vuejs.org/

Apa itu Vue.js?

cloud connect

- *an open-source front end JavaScript framework for building user interfaces and single-page applications.*
- *It was created by Evan You after working for Google using AngularJS in a number of projects.*
- *Vue was first released the following February, in 2014.*
- *Vue is generally used to create single-page apps that **run on the client** but can be used to create full-stack app by making HTTP request to backend server.*
- *Vue.js is a framework for **building client-side applications***
- *Vue can run on server side by using SSR (Server Side Rendering) Framework like Nuxt.js*

cloud connect

Vue.js **Versi 3**

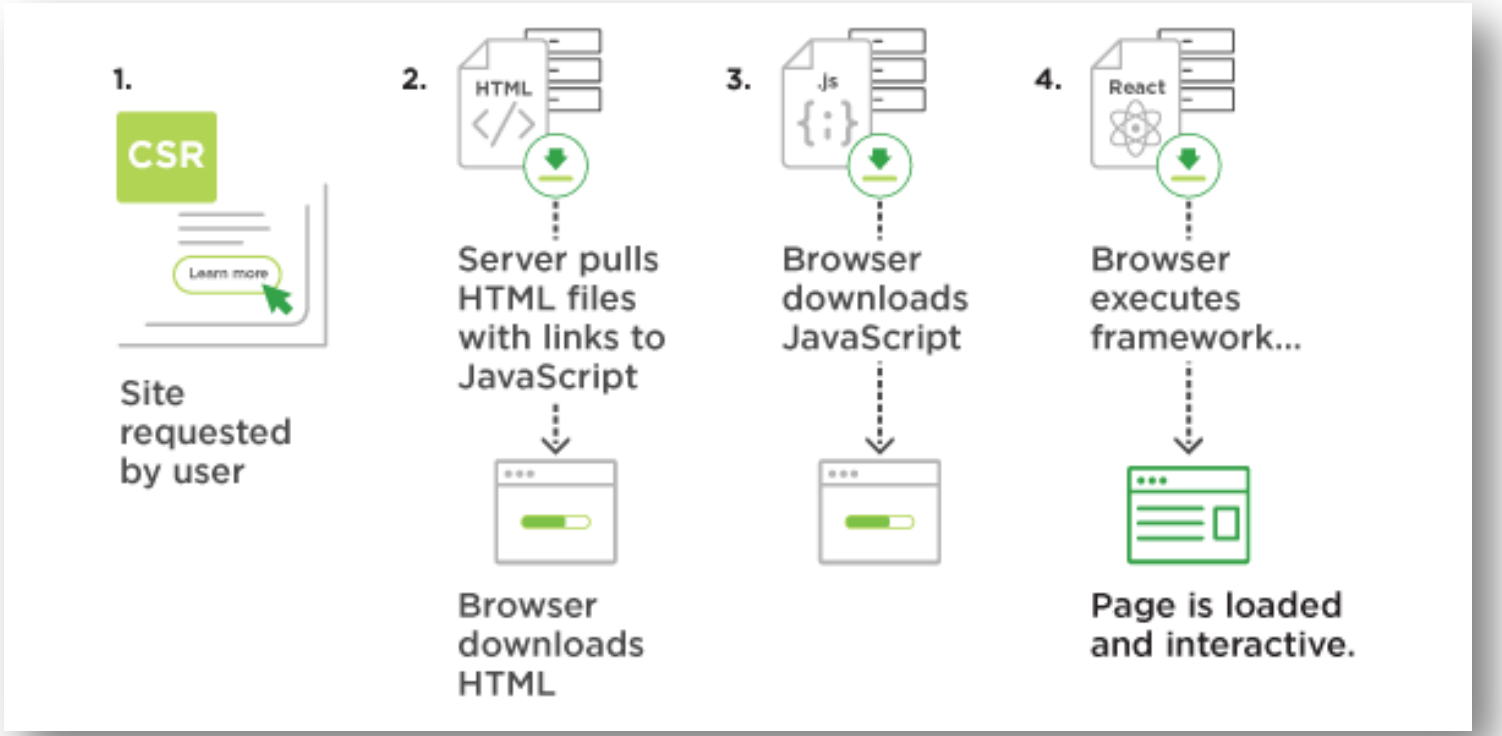
The Progressive JavaScript Framework

An approachable, performant and versatile framework for building web user interfaces.

Apa itu *Client Side Rendering*?

cloud connect

Rendering content in the browser using JavaScript. Instead of getting all of the content from the HTML document itself, you are getting a bare-bones HTML document with a JavaScript file that will render the rest of the site using the browser.



Vue - Single-File Components (SFC)

cloud connect

- *also known as *.vue files,*
- *A Vue SFC, as the name suggests, encapsulates the:*
  - *component's logic (JavaScript),*
  - *template (HTML), and*
  - *styles (CSS) in a single file*

Vue Components

cloud connect

- *Vue components can be authored in two different API styles:*
  - *Options API*
    - *define a component's logic using an object of options such as **data**, **methods**, and **mounted**. Properties defined by options are exposed on {**this**} inside functions*
    - *centered around the concept of a **"component instance"***
    - *aligns better with a class-based mental model for users coming from OOP language backgrounds.*
    - *It is also more **beginner-friendly** by abstracting away the reactivity details and enforcing code organization via option groups.*
  - *Composition API.*
    - *define a component's logic using imported API functions. In SFCs, Composition API is typically used with **<script setup>**.*
    - *centered around declaring reactive state variables directly in a function scope, and composing state from multiple functions together to handle complexity.*

Basic layout of an Options API

cloud connect

```
<template>

</template>

<script>
   export default {

   }
</script>

<style>

</style>
```

Vue page includes:

- a <template> for markup
- a <script> for javascript scripting
- a <style> for CSS styling

Apa itu *Vue Router*?

cloud connect

- *Vue Router is the official router for Vue.js. It deeply integrates with Vue.js core to make building Single Page Applications with Vue.js a breeze. Features include:*
  - Dynamic Route Matching
  - Routes' Matching Syntax
  - Named Routes
  - Nested Routes
  - Programmatic Navigation
  - Named Views
  - Redirect and Alias
  - Passing Props to Route Components
  - Active links
  - Different History modes

https://router.vuejs.org/

cloud connect

Latihan 9 – Vue.js