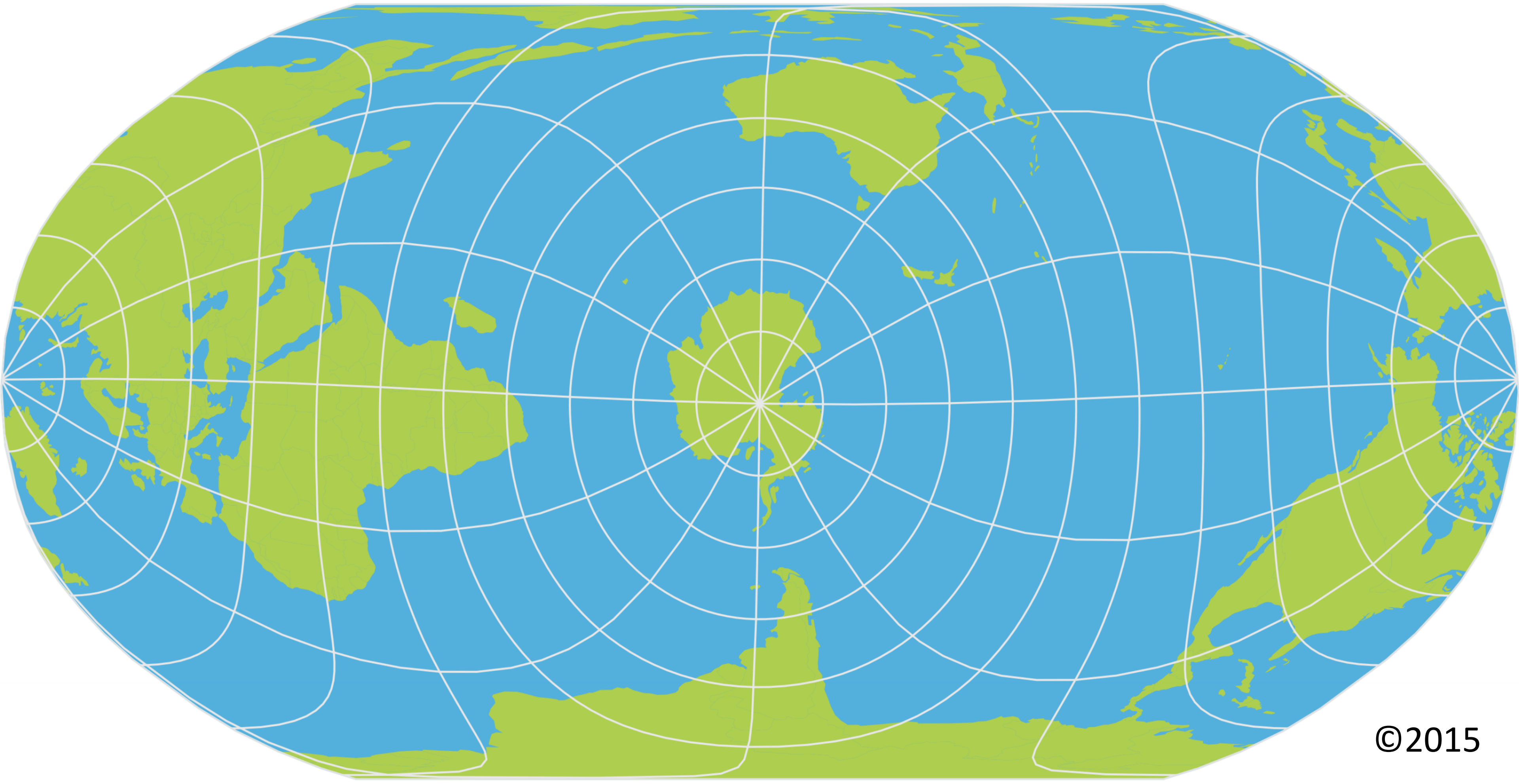
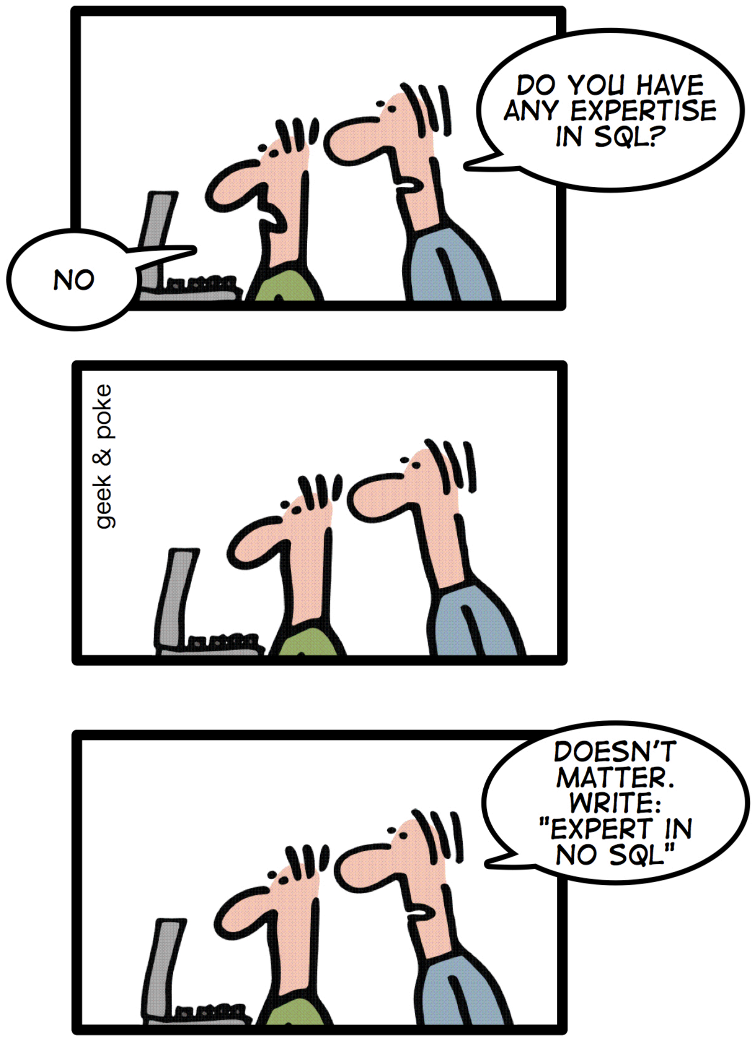
COS 787– NoSQL

V Rautenbach



HOW TO WRITE A CV


DO YOU HAVE ANY EXPERTISE IN SQL?
NO
geek & poke
DOESN'T MATTER. WRITE: "EXPERT IN NO SQL"

Leverage the NoSQL boom

What is NoSQL?

- https://www.youtube.com/watch?v=BgQFJ_UNIgw

What is NoSQL?

- NoSQL is defined as a new generation of database systems which have at least some of the following properties:
  - The underlying data model is **not relational**.
  - The systems are designed from the start as **horizontally and vertically scalable**.
  - The system is **open-source**.
  - The system **is schema-free** or has only gentle schema restrictions.
  - Because of the **distributed architecture**, the system supports a simple data replication method.
  - The system provides a straight forward API.
  - The system generally uses a different consistency model, but **not ACID**.

NoSQL

- Five situations in which data may be more suitable for a NoSQL system:
  - Various tables with lots of columns, each of which is only used by a few rows
  - Attribute tables
  - Numerous many-to-many relationships
  - Tree-like characteristics
  - Requiring frequent schema changes

NoSQL prohibits structured query language (SQL)

- True
- False

Types of NoSQL dbs

- Key/value:
  - Store data in **key/value pairs**
  - Very efficient for performance and highly scalable, but difficult to query and to implement real-world problems
- Column:
  - Store data in **tabular structures**, but columns may vary in time and each row may have only a subset of the columns
- Document oriented:
  - Like key/value, but lets you store **more values for a key**
  - A document value could be for example an xml or json fragment
  - This is a nice paradigm for programmers as it becomes easy, especially with script languages, to implement a one-to-one mapping relation between the code objects and the objects (documents) in the system
- Graph:
  - Stores objects and relationships in **nodes and edges** of a graph.
  - For situations that fit this model, like hierarchical data, this solution could be faster than the other ones.

Consistency, availability and partition tolerance

- Many NoSQL systems are designed with the CAP theorem in mind.
  - **Consistency** means transactions are ACID (Atomic, Consistent, Isolated, Durable). All nodes see the same data at the same time.
  - **Availability** means all data must be available and all transactions must come to an end (within a reasonable time). If a node fails, the other nodes continue.
  - **Partition-tolerance** is the fault-tolerance between individual components in a distributed environment. This means the system continues even if messages between components are lost.

Sharding

- Sharding is a type of **database partitioning** that separates very large databases the into smaller, faster, more easily managed parts called data shards.
- The word shard means a small part of a whole.

Each field in a MongoDB database can be indexed

- What does indexing a database help to achieve?
- Describe at a conceptual level how indexing is implemented in MongoDB.
- Indexes can be on a single field or composite across more than one field. Describe how a data is accessed via a composite index.
- Discuss the reasoning you would follow when deciding which fields to index in a database. How would you choose the single and composite indexes and why wouldn’t you just index everything?

NoSQL databases such as MongoDB are often said to be schemaless

- Describe what is meant by a schema in terms of a relational database
- Taking a MongoDB document store as an example, describe what it
- means to say that NoSQL databases do not use a schema based
- design.
- Name two advantages of a schemaless database design
- Name two disadvantages of a schemaless database design

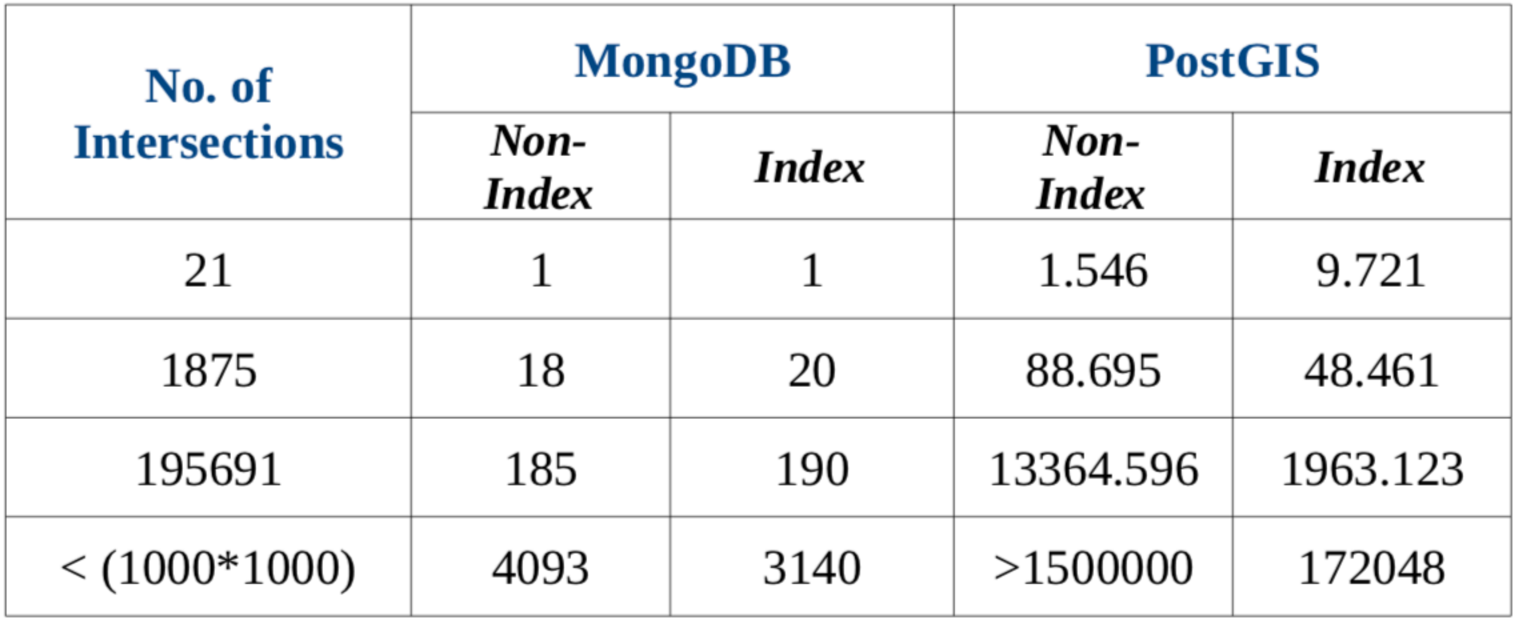
Indexing

- https://www.youtube.com/watch?v=tSgPhxZdhLk

NoSQL and spatial data

- When do you use NoSQL vs PostGIS?

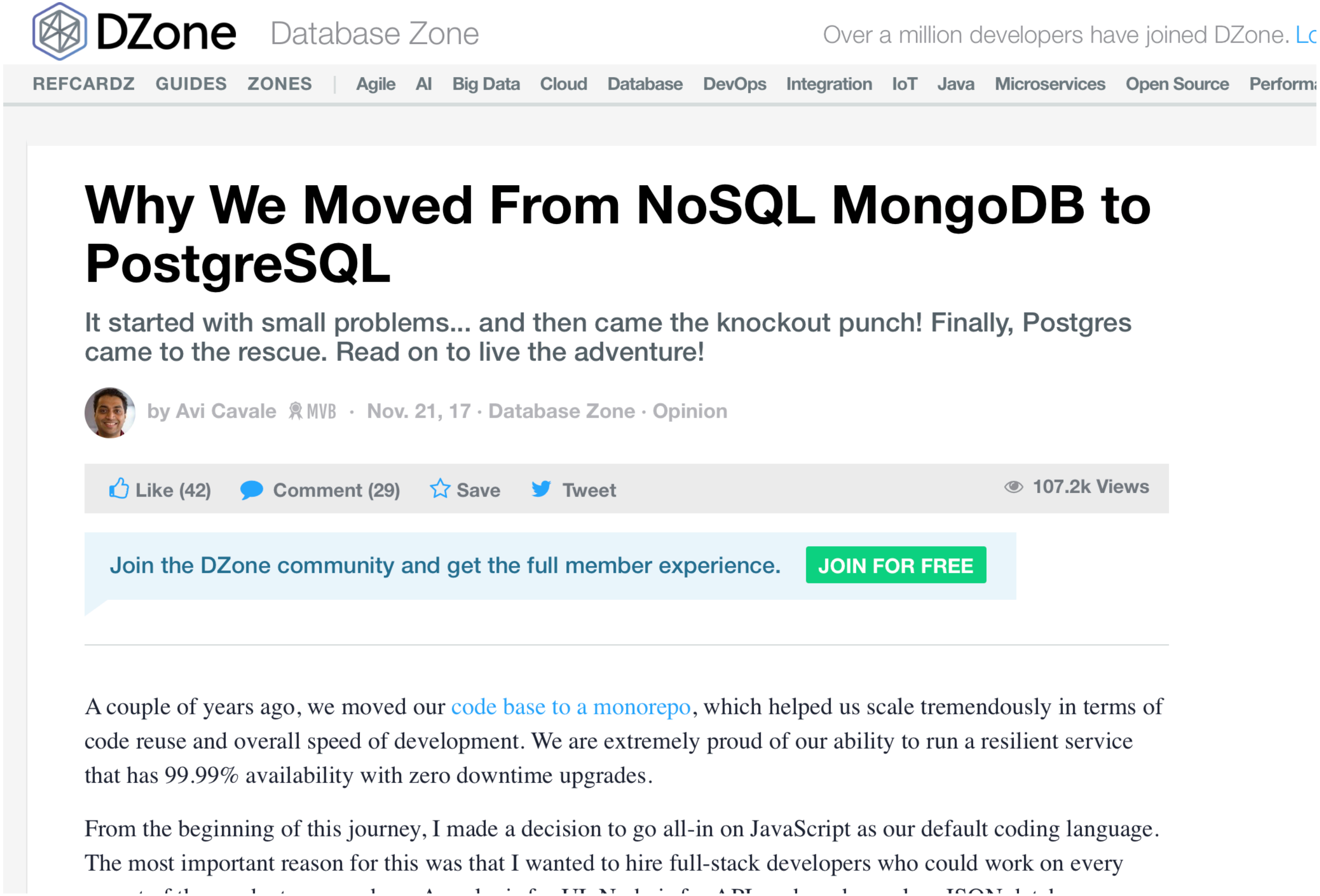
MongoDB vs PostGIS



No. of Intersections	MongoDB		PostGIS	
	Non-Index	Index	Non-Index	Index
21	1	1	1.546	9.721
1875	18	20	88.695	48.461
195691	185	190	13364.596	1963.123
< (1000*1000)	4093	3140	>1500000	172048

MongoDB vs PostGIS

- The basic functionality is available in both, and you should decide considering all other factors:
  - Is your data well structured, and with few variations within this structure? (more on SQL side)
  - Do you need ACID (atomicity, consistency, isolation, durability) guarantees? (more on SQL side)
  - Do you need huge horizontal scaling? (more on MongoDB side)
  - Do you need to query several tables at once (and join)? (SQL side)
  - Do you feel more comfortable with JavaScript and JSON than in any other languages? (MonboDB +)
  - (etc., etc.)


DZone Database Zone
Over a million developers have joined DZone. Lo
REFCARDZ GUIDES ZONES | Agile AI Big Data Cloud Database DevOps Integration IoT Java Microservices Open Source Perform

Why We Moved From NoSQL MongoDB to PostgreSQL

It started with small problems... and then came the knockout punch! Finally, Postgres came to the rescue. Read on to live the adventure!

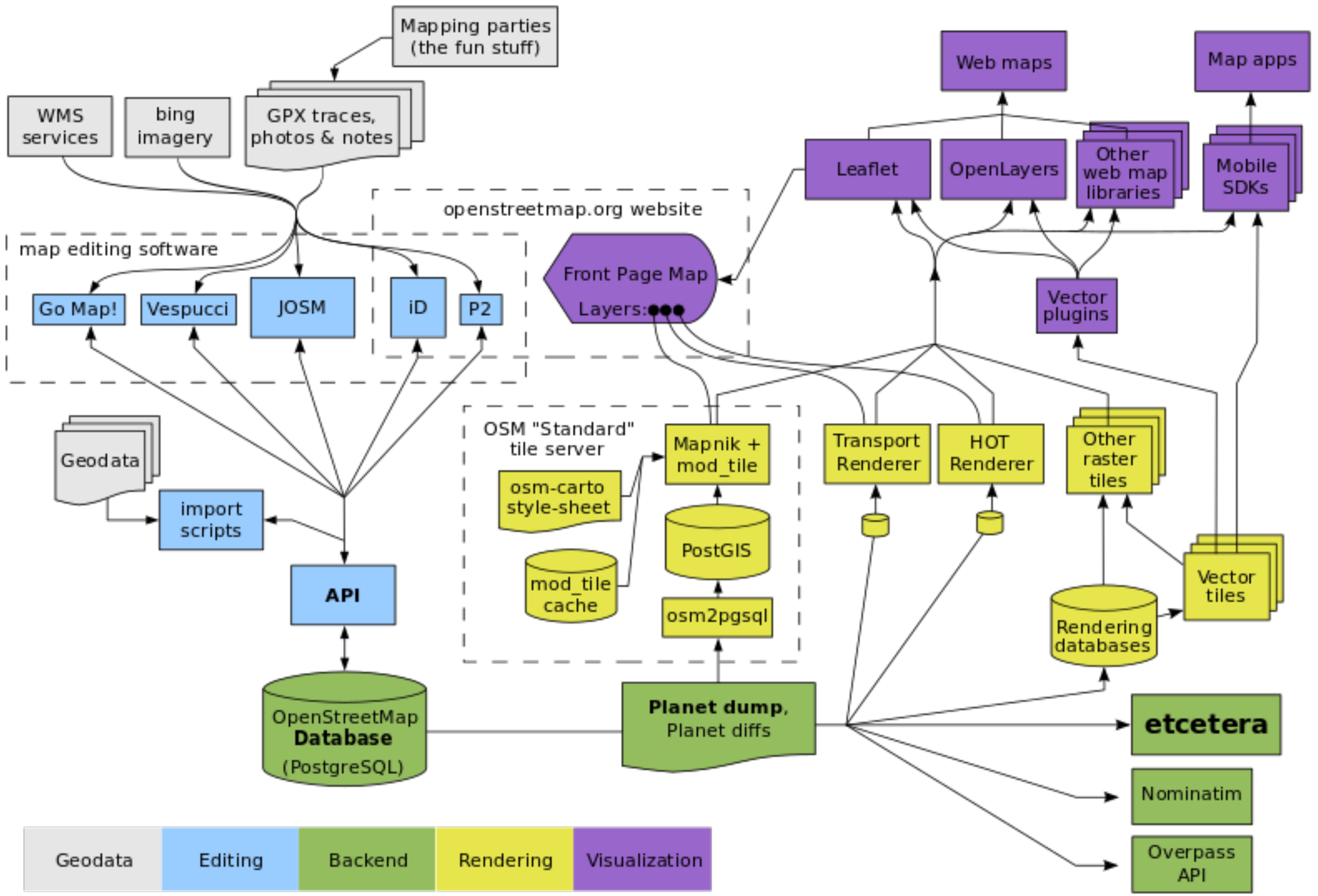
by Avi Cavale MVB · Nov. 21, 17 · Database Zone · Opinion

Like (42) Comment (29) Save Tweet 107.2k Views
Join the DZone community and get the full member experience. JOIN FOR FREE

A couple of years ago, we moved our code base to a monorepo, which helped us scale tremendously in terms of code reuse and overall speed of development. We are extremely proud of our ability to run a resilient service that has 99.99% availability with zero downtime upgrades.

From the beginning of this journey, I made a decision to go all-in on JavaScript as our default coding language. The most important reason for this was that I wanted to hire full-stack developers who could work on every

Source: https://dzone.com/articles/why-we-moved-from-nosql-mongodb-to-postgresql



Mapping parties (the fun stuff)
Web maps
Map apps
WMS services
bing imagery
GPX traces, photos & notes
Leaflet
OpenLayers
Other web map libraries
Mobile SDKs
openstreetmap.org website
map editing software
Front Page Map
Layers:●●●
Go Map!
Vespucci
JOSM
iD
P2
Vector plugins
Geodata
import scripts
OSM "Standard" tile server
Mapnik + mod_tile
osm-carto style-sheet
Transport Renderer
HOT Renderer
Other raster tiles
PostGIS
mod_tile cache
API
osm2pgsql
Vector tiles
Rendering databases
OpenStreetMap Database (PostgreSQL)
Planet dump, Planet diffs
etcetera
Nominatim
Overpass API
Geodata
Editing
Backend
Rendering
Visualization

victoria.rautenbach@up.ac.za