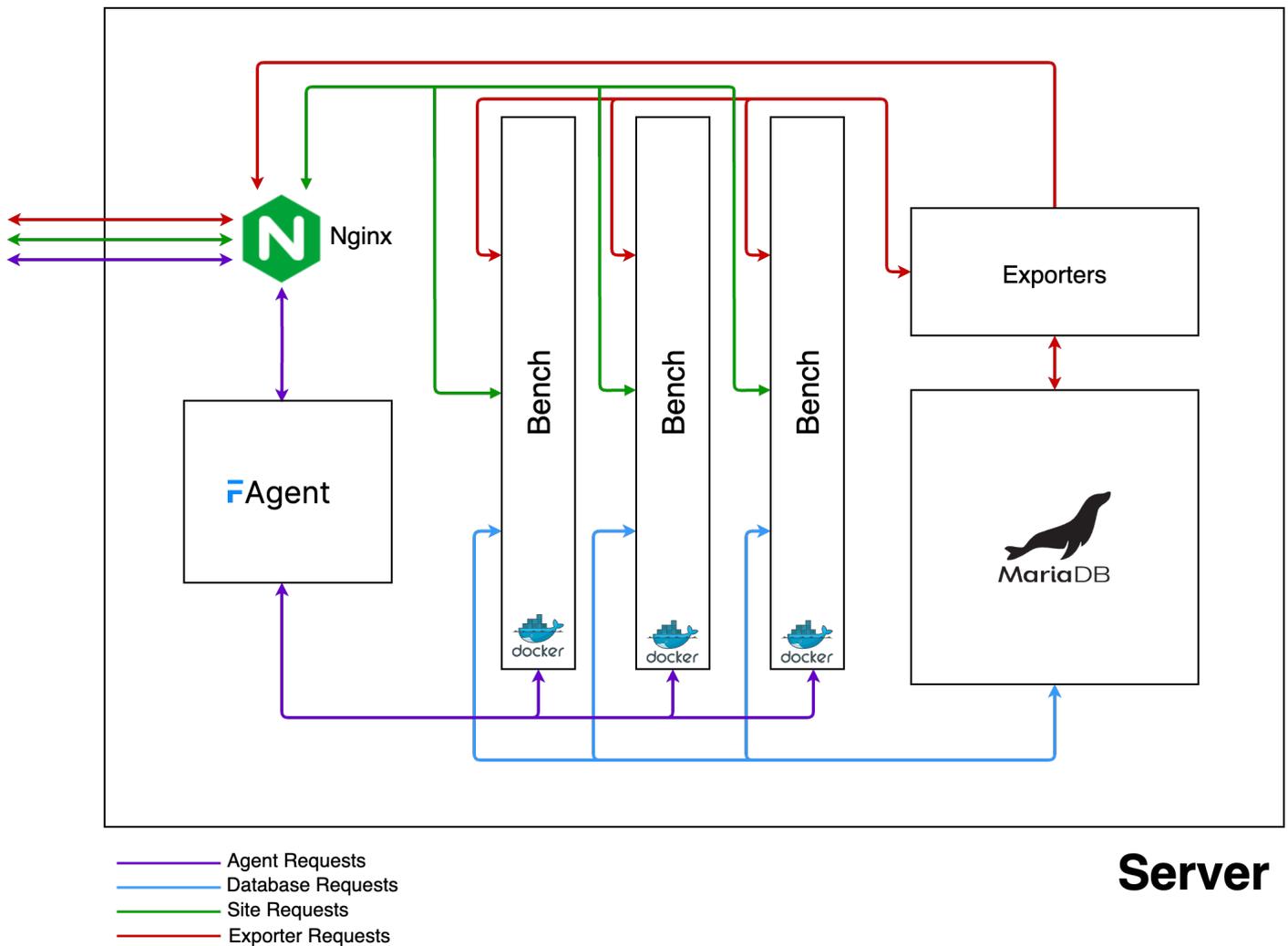


Architecture

- [High Level Diagram Architecture](#)
- [Low Level Diagram Architecture](#)

High Level Diagram Architecture

1. **Infrastructure Architecture HLL:** An instance of tibERbu runs on an environment called a bench. It exposes all functionalities for use by client requests, application function calls or automated tasks. A **bench** can sit on a *physical server*, *a virtual machine* or an *orchestration cluster*. **Frappe Cloud(FAgent)** is an orchestration layer that can accommodate multiple benches and is the basis for setting up the core components of tibERbu EMR(database, redis-cache, and the main engine Frappe Framework). Each bench will have a list of applications ("apps") that run as docker containers inside the bench. An app is usually hosted on GitHub/GitLab and linked to Bench using Frappe Cloud. Frappe Cloud comes with automated CD pipelines that automatically detect updates and provide a user with actions to update and recreate containers, thereby updating the apps. A **site** is where users interact with the ERP/EMR functionalities through a browser like Google Chrome, Safari, Firefox or any other.



Server

1. **Application Architecture** An "app" provides functionalities that enable functions ranging from saving data, retrieving data and displaying it on an user interface for actions like editing or deletion to complex data exchange functions like REST, webhooks and socket functionalities. Frappe Framework wraps best practice functionalities to act on data or send data via email. Frappe follows a document based architecture which means all app data is presented in JSON format and transmitted as such.

Sites

pcn01.county.com

l1.county.com

l2.county.com

l3.county.com

Apps

frappe

erpnext

Custom_Apps

CORE RUNTIMES (NODE, WERKZEUG, REDIS, BACKGROUND SERVICES)

PYTHON VIRTUAL ENVIRONMENT

MARIADB/POSTGRES CLIENT

INSIGHTS / ANALYTICS

Low Level Diagram Architecture

To recap main building blocks of tibERbu HealthNet stack are:

- **bench** - The environment that makes all functionalities possible. Adding sites, installing apps, performing runtimes, etc.
- **app** - It defines specific functionalities such as core services, Document Models(doctypes)(ORM) schemas, specific REST interfaces and hooks (event triggers)
- **site** - This enables the user to interact with system functionalities. Each site is its own tenant with full end-to-end capabilities.(UI - program logic - cache - database)

***then finally **The Cloud**: It is an orchestration layer that hosts one or more benches to ensure smooth DevOPs, continuous delivery and ultimately contribute to a great user experience!

