

fans🔒

Technical Documentation



I. Introduction

Fanso backend design defines the goals of the architecture, architectural styles, and components that have been implemented. The document is used to capture the architecture and design decisions made from the conceptual idea to its implementation.

II. System overview

Fanso system is comprised of various modules / micro-services. Each service is responsible for providing APIs to help users execute use-cases in a specific domain.

System and service	Description
Auth	Service provides APIs to execute all use-cases related to authentication and authorization such as Sign In, Sign Out, Access Token Verification and User's Roles and Permissions and etc.
User	Service provides APIs to execute all use-cases related to user domain such as Get, Edit Profile and etc
Performer	Service provides APIs to execute all use-cases related to performer domain such as Get, Edit Profile
Performer-assets	Service provides APIs to execute all use-cases related to assets such as videos, photos, albums
File	Service provides APIs to execute all use-cases related to file storage such as convert file, create thumbnails....
Payment	Service provides APIs to execute all use-cases related to payment such as Stripe payment, payment history...
Payout	Service provides APIs to execute all use-cases related to payout request
Media	Service provides APIs to execute all use-cases related to files such as upload, convert file to mp4, resize image, etc...

III. Technologies

Technologies are used to build Fanso system

1. API / Backend

- > Language / framework: Javascript (NodeJS)
- > Data storages: MongoDB, Redis
- > Queue: Redis

- > Cache: Redis
- > Video processor: FFMPEG
- > Websocket: Socket.io
- > File secure: Nginx with http_auth_module

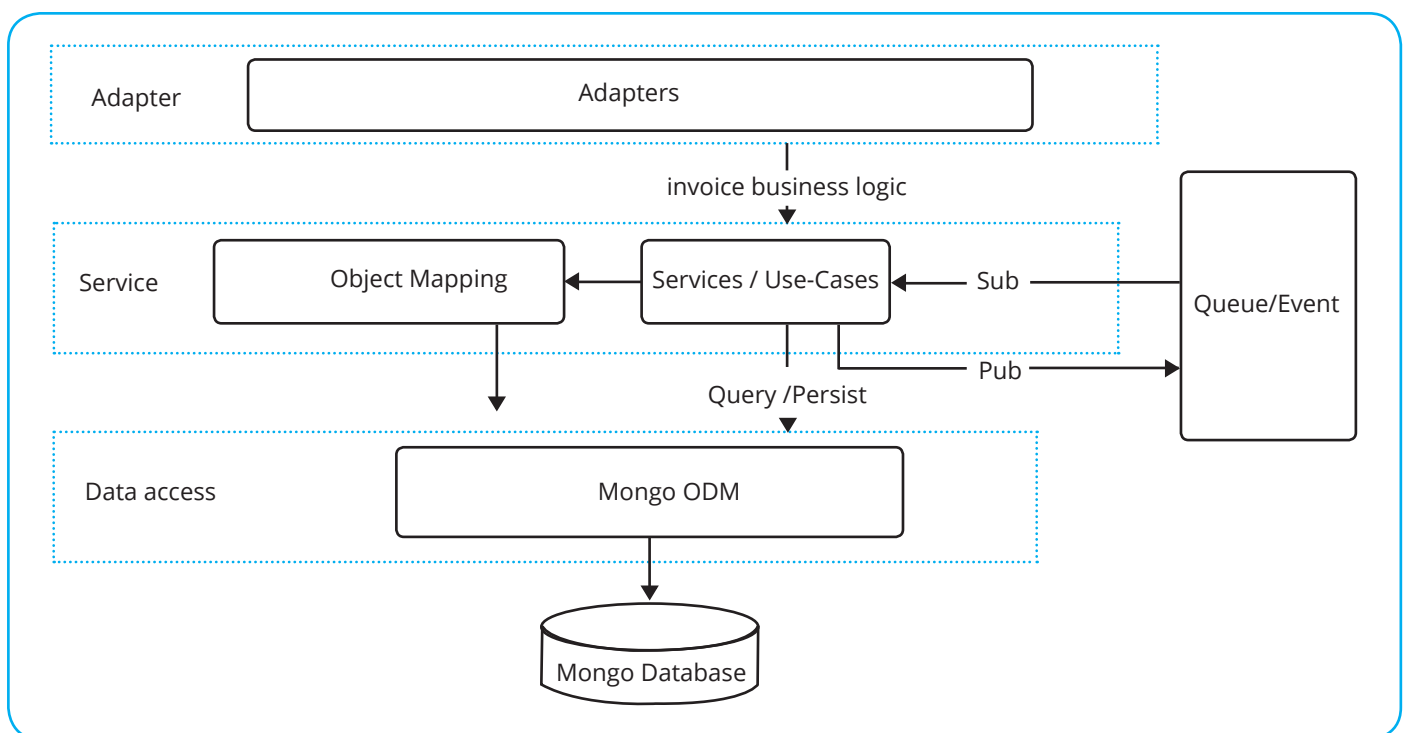
2. Frontend

- > Language / framework: NextJS
- > CSS framework: Ant design

IV. Service Architecture

This architecture style is applied for services that have simple domain logic and mainly focus on serving data querying for specific views and users.

The picture below illustrates all main components in this architecture and the collaboration among them:



Components	Descriptions
Adapters	API Layer is responsible for exposing all APIs to the outside world via protocols such as REST, gRPC, or GraphQL. This layer handles back compatibilities with frontend applications
Object Mapping	Transfer Data Object models between DB and views
Event publisher	Event Publisher is responsible for publishing event, message arose from use-cases execution
Mongo ODM	Provides a straight-forward, schema-based solution to model application data. Use Mongoose plugin

V. Secure your data

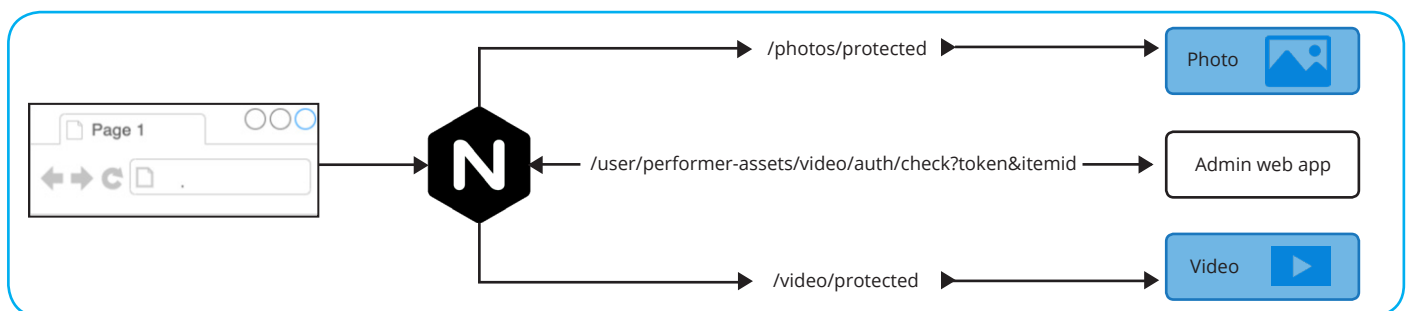
Technologies are used to build Fanso system

Password

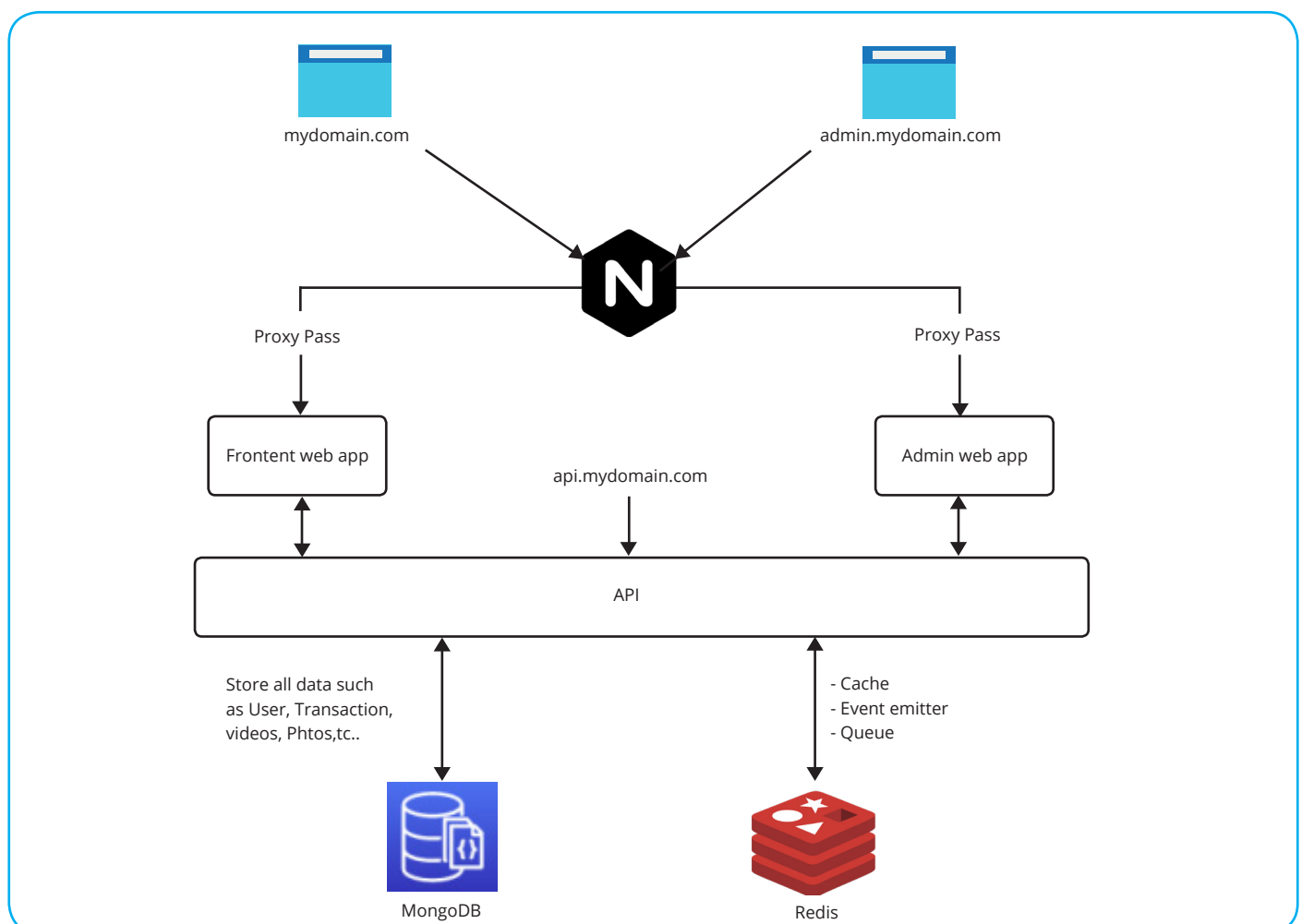
- > pbkdf2 method (Password-Based Key Derivation Function 2) with salt is generated randomly for each user Secret key might be provided from process environment

Files

- > Fanso use [nginx_http_auth_module](#) to verify request access to protected files and allow / prevent access



High level design Architecture



I. Server requirement

Fanso supports all platforms: Windows, MacOS, Linux.

Fanso needs a VPS server with at least:

- 2GB of RAM - recommend 4GB
- 40GB of HDD
- 1 CPU core - recommend 2 cores at least
- 3 domains / subdomains
 - > api.[your-domain] and point to server IP address
 - > admin.[your-domain] and point to server IP address
 - > [your-domain] and point to server IP address

II. Software requirements

Fanso architecture needs these softwares

- [NodeJS](#) v12.x
 - >To install please download NodeJS [here](#)
- [MongoDB](#) >= v3.6
 - >To install please [download mongoDB](#)
- [Redis server](#) >= v2.8
 - >To install please [download](#) and setup redis
- [FFMPEG](#)
- [Nginx](#) >= v1.3 with http_auth_module enabled
 - [PM2](#) is a daemon process manager that will help you manage and keep your application online 24/7
- [Yarn](#) or npm to manage nodeJS package

Ensure all softwares above are running before setup source code

Tech FAQ

1. How does Fanso store asset files such as videos and images?

Answer: Fanso supports Local storage (single machine) for now. We use nginx [http_auth_module](#) to protect assets.

2. Does Fanso support S3 (Simple Storage Service) services like AWS S3 or Digital Ocean Space?

Answer: Fanso supports local storage for now. If need S3 service you can customize our File module or [contact for customization](#)

3. Does Fanso support FTP file server?

Answer: Fanso supports local storage for now. If need FTP service you can customize our File module or [contact for customization](#)

4. Can we deploy Fanso to AWS EC2?

Answer: Yes, Fanso is able to be deployed to any VPS server, include AWS EC2

5. Does Fanso support Kubernetes?

Answer: We have not supported the application container image in our script yet. So you have to create an image e.g. with Docker and deploy an application to Kubernetes. Or [contact us for a customization](#).

6. Does Fanso support Docker?

Answer: For now Fanso does not provide Docker images yet. We will provide it in future releases.

7. Is Fanso using CI/CD in our dev process?

Answer: we are using Jenkins for this purpose

8. How can we deliver the update efficiently?

Answer: We provide full source code, so you can do implementation if you want. Or we can provide Docker image for your customizations



Get Started With fanso



Live Demo



Pricing



FAQ



+66 93 353 5223



+1 408 909 5136



www.fanso.io



Schedule a
15mins free call with CEO of Fanso

