**Documentation Technique - Meteor Projet**

**1. Introduction**

**Présentation de l'application**

**Meteor Projet** est un outil de gestion simplifié de projet destiné aux chefs de projet, managers et participants aux projets. Son objectif est de faciliter la gestion des tâches, la revue de projet, la gestion des risques et l'export de présentations PowerPoint.

**Public cible**

L'application s'adresse principalement aux :

* Chefs de projet
* Managers
* Participants aux projets

**2. Architecture de l’application**

**Stack technique**

Meteor Projet repose sur la technologie suivante :

* **Frontend** : VITE + TypeScript
* **Base de données** : Supabase
* **Déploiement** : Conteneurisation via Docker

**Hébergement**

L’application est hébergée **On-Premise** sur une infrastructure interne.

**Déploiement en conteneurs**

L’architecture comprend :

* Un conteneur pour l'application (VITE + TypeScript)
* Plusieurs conteneurs pour **Supabase**

**3. Installation et Configuration**

**Prérequis techniques**

* Docker et Docker Compose installés sur l’hôte (éventuellement Portainer)
* Accès réseau aux serveurs hébergeant Supabase
* Configuration d’un fichier d’environnement

**Installation Supabase**

**En local :**

1. Cloner le dépôt du projet
2. Lancer la commande docker-compose up -d
3. Accéder à l'application via l'URL configurée

**En production :**

1. Déploiement identique via Docker avec des configurations spécifiques adaptées à l’environnement

**Fichiers de configuration**

* docker-compose.yml : Définit la configuration des conteneurs
* .env : Contient les variables d’environnement nécessaires

Documentation complète disponible ici : [Self-Hosting with Docker | Supabase Docs](https://supabase.com/docs/guides/self-hosting/docker)

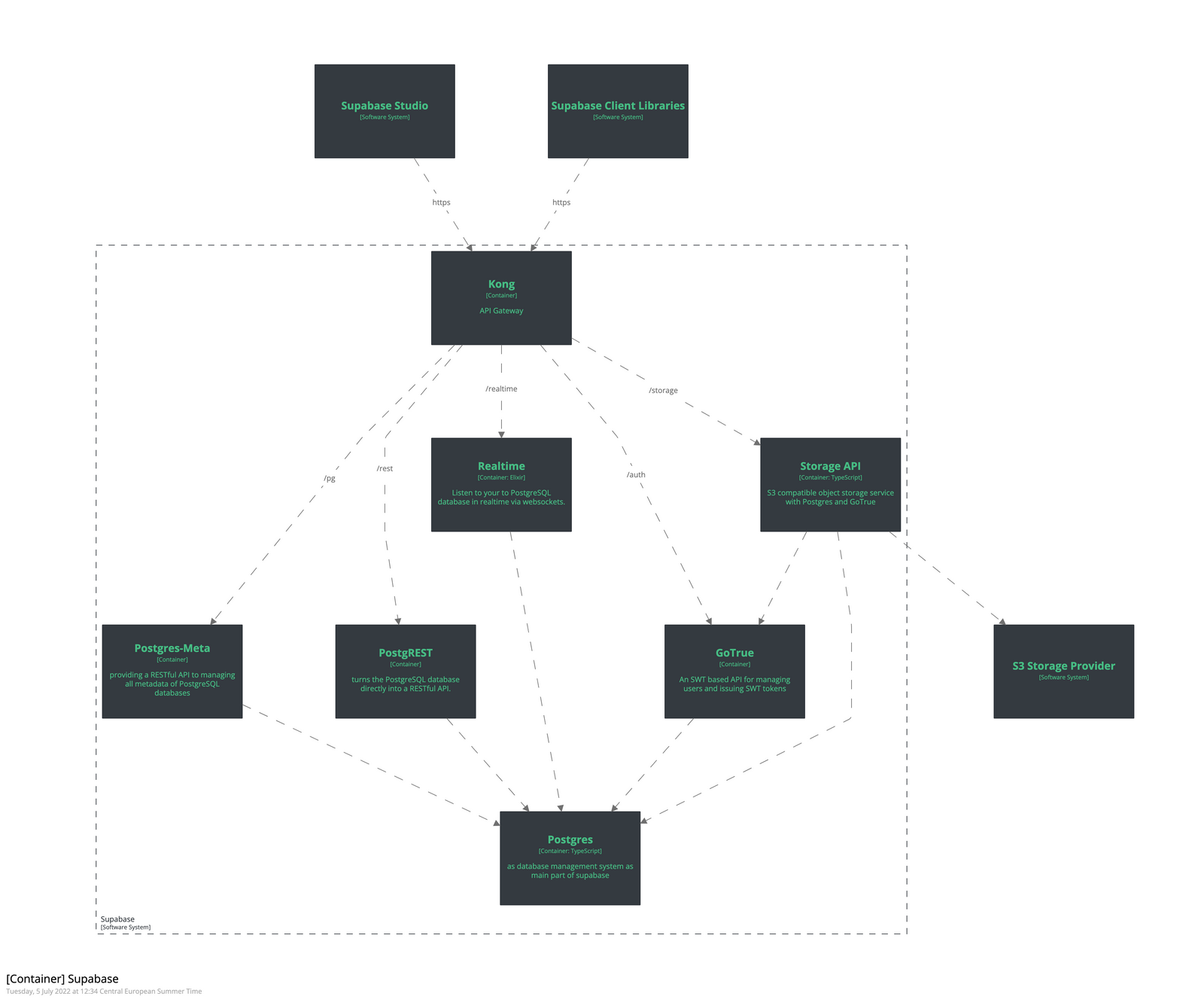
**Fichiers de données**

* Données pérennes sont stockés dans des volumes spécifiques, montés depuis l'hôte.

**DNS :**

* L’accès aux données se fait via une API REST, accessible sur <http://apimeteor>

**Architecture docker de supabase :**



**Installation Meteor Projet**

**En production :**

1. Déploiement vie une stack depuis portainer

image: spottier88/project-meteor:latest

container\_name: meteor-app

**Accès :** [**http://meteor**](http://meteor)

**Installation Nginx**

Un container NGINX est également déployer sur la machine hote de docker afin de limier l’ouverture des ports.

[**http://meteor**](http://meteor) **renvoie vers le container meteor-app**

[**http://apimeteor**](http://apimeteor) **renvoi vers le container kong de supabase (port 8000)**

**4. Fonctionnalités principales**

**Fonctionnalités disponibles**

* **Gestion de projet** : Création, suivi et modification de projets
* **Revue de projet** : Analyse et reporting de l’avancement
* **Gestion des tâches** : Attribution, suivi et validation des tâches
* **Gestion des risques** : Identification et évaluation des risques
* **Export PowerPoint** : Génération de présentations basées sur les données du projet

**5. Authentification et intégrations**

**Authentification et autorisation**

* **Méthodes disponibles** :
  + Login/Mot de passe
  + Magic Link (connexion sans mot de passe via email)

**Intégrations avec d'autres services**

* Aucune intégration externe n’a été mise en place pour le moment.

**6. Base de données**

**Type de base de données**

* **Technologie** : Supabase (PostgreSQL)

**Schéma de base de données**

Une image contenant texte, capture d’écran

Description générée automatiquement

**Gestion des migrations de données**

* Les mises à jour et modifications de la base de données sont gérées via des scripts SQL de migration.

**7. Sécurité**

**Gestion des accès**

* Utilisation de l’authentification par login/mot de passe ou Magic Link
* Sécurisation des accès aux projets en fonction des rôles utilisateurs

**Sécurisation des données**

* Les pratiques de sécurisation sont mises en place selon les recommandations de Supabase
* Les permissions et accès sont configurés pour éviter les fuites de données, via des politiques RLS dans supabase

**8. Déploiement et Maintenance**

**Déploiement**

* Utilisation de **stacks Docker** pour l’ensemble des composants
* Hébergement On-Premise avec monitoring intégré

**Monitoring et logs**

* Supervision et logs fournis par **Supabase**

**Mise à jour des versions**

* Les nouvelles versions sont déployées via des mises à jour des **stacks Docker**

**Sauvegarde et restauration**

* Un fichier de script est fourni pour sauvegarder la base de données de l’application, ainsi que les données d’environnement et de config.
  + Disponible sour /home/dockadm/scripts/supabase\_backup.sh
  + Executé tous les soirs à 2h du matin (crontab)
  + Stocke de manière incrémentale (7 jours) les sauvegardes de la base sous format tar.gz

**9. Support et Dépannage**

**Documentation utilisateur**

* Une documentation est disponible pour l’administration de l’application

**Problèmes courants et solutions**

* Les erreurs courantes et leurs résolutions seront documentées dans un guide spécifique

**Support technique**

* Gestion du support via **GLPI**