

Smart VMC

1.0

Generated on Fri Jan 16 2026 14:56:18 for Smart VMC by Doxygen 1.15.0

Fri Jan 16 2026 14:56:18

Chapter 1

Directory Hierarchy

1.1 Directories

components	??
dashboard	??
ControlPanel.tsx	??
SensorRow.tsx	??
StatusDisplay.tsx	??
ui	??
AppButton.tsx	??
AppInput.tsx	??
Card.tsx	??
ConfigurationPanel.tsx	??
DashboardPanel.tsx	??
SetupPanel.tsx	??
TabHeader.tsx	??
dashboard	??
ControlPanel.tsx	??
SensorRow.tsx	??
StatusDisplay.tsx	??
firmware	??
src	??
main.cpp	??
main.hpp	??
hooks	??
useBLE.ts	??
useWindowApi.ts	??
mobile	??
components	??
dashboard	??
ControlPanel.tsx	??
SensorRow.tsx	??
StatusDisplay.tsx	??
ui	??
AppButton.tsx	??
AppInput.tsx	??
Card.tsx	??
ConfigurationPanel.tsx	??
DashboardPanel.tsx	??

SetupPanel.tsx	??
TabHeader.tsx	??
hooks	??
useBLE.ts	??
useWindowApi.ts	??
services	??
NotificationService.ts	??
.eslintrc.js	??
.prettierrc.js	??
App.tsx	??
babel.config.js	??
index.js	??
jest.config.js	??
metro.config.js	??
services	??
NotificationService.ts	??
src	??
main.cpp	??
main.hpp	??
ui	??
AppButton.tsx	??
AppInput.tsx	??
Card.tsx	??

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

AppButtonProps	??
BLECharacteristicCallbacks	
ConfigCallbacks	??
ConfigCallbacks	??
CardProps	??
ConfigurationPanelProps	??
ControlPanelProps	??
DashboardPanelProps	??
NotificationService	??
SensorRowProps	??
SetupPanelProps	??
StatusDisplayProps	??
TabHeaderProps	??
TextInputProps	
AppInputProps	??
WindowState	??

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

AppButtonProps	
Propriétés du bouton	??
AppInputProps	
Propriétés du champ de saisie	??
CardProps	
Propriétés du composant carte	??
ConfigCallbacks	
Traite les configurations WiFi et de localisation	??
ConfigurationPanelProps	
Propriétés du panneau de configuration	??
ControlPanelProps	
Propriétés du panneau de contrôle	??
DashboardPanelProps	
Propriétés du tableau de bord	??
NotificationService	??
SensorRowProps	
Propriétés de la ligne de capteurs	??
SetupPanelProps	
Propriétés du panneau de configuration initiale	??
StatusDisplayProps	
Propriétés de l'affichage d'état	??
TabHeaderProps	
Propriétés de l'en-tête d'onglets	??
WindowState	
État de la fenêtre côté mobile	??

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

firmware/src/main.cpp	??
firmware/src/main.hpp	??
mobile/.eslintrc.js	??
mobile/.prettierrc.js	??
mobile/App.tsx	
Composant principal de l'application mobile Smart VMC	??
mobile/babel.config.js	??
mobile/index.js	??
mobile/jest.config.js	??
mobile/metro.config.js	??
mobile/components/ConfigurationPanel.tsx	
Panneau de configuration des seuils de température et pollution	??
mobile/components/DashboardPanel.tsx	
Tableau de bord affichant l'état, les capteurs et les commandes de la VMC	??
mobile/components/SetupPanel.tsx	
Panneau de configuration initiale avec saisie WiFi et localisation	??
mobile/components/TabHeader.tsx	
Barre d'onglets principale	??
mobile/components/dashboard/ControlPanel.tsx	
Panneau de contrôle de la VMC (mode auto, commandes manuelles, slider)	??
mobile/components/dashboard/SensorRow.tsx	
Ligne d'affichage des valeurs de capteurs (température et AQI)	??
mobile/components/dashboard/StatusDisplay.tsx	
Affichage de l'état global de la VMC (ouverte/fermée/verrouillée)	??
mobile/components/ui/AppButton.tsx	
Bouton applicatif stylé (primaire, danger, outline) avec état de chargement	??
mobile/components/ui/AppInput.tsx	
Champ de saisie stylé avec placeholder grisé	??
mobile/components/ui/Card.tsx	
Conteneur carte avec ombre et arrondis	??
mobile/hooks/useBLE.ts	
Hook React pour la communication Bluetooth avec l'ESP32	??
mobile/hooks/useWindowApi.ts	
Hook React pour communiquer avec l'API smart-vmc	??
mobile/services/NotificationService.ts	??

Chapter 5

Directory Documentation

5.1 mobile/components Directory Reference

Directories

- directory [dashboard](#)
- directory [ui](#)

Files

- file [ConfigurationPanel.tsx](#)
Panneau de configuration des seuils de température et pollution.
- file [DashboardPanel.tsx](#)
Tableau de bord affichant l'état, les capteurs et les commandes de la VMC.
- file [SetupPanel.tsx](#)
Panneau de configuration initiale avec saisie WiFi et localisation.
- file [TabHeader.tsx](#)
Barre d'onglets principale.

5.2 mobile/components/dashboard Directory Reference

Files

- file [ControlPanel.tsx](#)
Panneau de contrôle de la VMC (mode auto, commandes manuelles, slider).
- file [SensorRow.tsx](#)
Ligne d'affichage des valeurs de capteurs (température et AQI).
- file [StatusDisplay.tsx](#)
Affichage de l'état global de la VMC (ouverte/fermée/verrouillée).

5.3 firmware Directory Reference

Directories

- directory [src](#)

5.4 mobile/hooks Directory Reference

Files

- file [useBLE.ts](#)
Hook React pour la communication Bluetooth avec l'ESP32.
- file [useWindowApi.ts](#)
Hook React pour communiquer avec l'API smart-vmc.

5.5 mobile Directory Reference

Directories

- directory [components](#)
- directory [hooks](#)
- directory [services](#)

Files

- file [.eslintrc.js](#)
- file [.prettierrc.js](#)
- file [App.tsx](#)
Composant principal de l'application mobile Smart VMC.
- file [babel.config.js](#)
- file [index.js](#)
- file [jest.config.js](#)
- file [metro.config.js](#)

5.6 mobile/services Directory Reference

Files

- file [NotificationService.ts](#)

5.7 firmware/src Directory Reference

Files

- file [main.cpp](#)
- file [main.hpp](#)

5.8 mobile/components/ui Directory Reference

Files

- file [AppButton.tsx](#)
Bouton applicatif stylé (primaire, danger, outline) avec état de chargement.
- file [AppInput.tsx](#)
Champ de saisie stylé avec placeholder grisé
- file [Card.tsx](#)
Conteneur carte avec ombre et arrondis.

Chapter 6

Class Documentation

6.1 AppButtonProps Interface Reference

Propriétés du bouton.

6.1.1 Detailed Description

Propriétés du bouton.

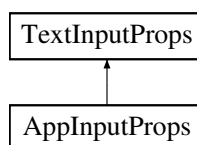
The documentation for this interface was generated from the following file:

- mobile/components/ui/[AppButton.tsx](#)

6.2 AppInputProps Interface Reference

Propriétés du champ de saisie.

Inheritance diagram for AppInputProps:



6.2.1 Detailed Description

Propriétés du champ de saisie.

The documentation for this interface was generated from the following file:

- mobile/components/ui/[AppInput.tsx](#)

6.3 CardProps Interface Reference

Propriétés du composant carte.

6.3.1 Detailed Description

Propriétés du composant carte.

The documentation for this interface was generated from the following file:

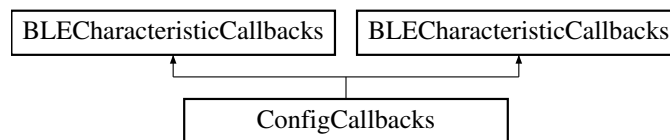
- [mobile/components/ui/Card.tsx](#)

6.4 ConfigCallbacks Class Reference

Traite les configurations WiFi et de localisation.

```
#include <main.hpp>
```

Inheritance diagram for ConfigCallbacks:



6.4.1 Detailed Description

Traite les configurations WiFi et de localisation.

The documentation for this class was generated from the following files:

- [firmware/src/main.cpp](#)
- [firmware/src/main.hpp](#)

6.5 ConfigurationPanelProps Interface Reference

Propriétés du panneau de configuration.

6.5.1 Detailed Description

Propriétés du panneau de configuration.

The documentation for this interface was generated from the following file:

- [mobile/components/ConfigurationPanel.tsx](#)

6.6 ControlPanelProps Interface Reference

Propriétés du panneau de contrôle.

6.6.1 Detailed Description

Propriétés du panneau de contrôle.

The documentation for this interface was generated from the following file:

- [mobile/components/dashboard/ControlPanel.tsx](#)

6.7 DashboardPanelProps Interface Reference

Propriétés du tableau de bord.

6.7.1 Detailed Description

Propriétés du tableau de bord.

The documentation for this interface was generated from the following file:

- [mobile/components/DashboardPanel.tsx](#)

6.8 NotificationService Class Reference

Public Member Functions

- `async` [requestPermission](#) ()
- `async` [createChannel](#) ()
- `async` [displayNotification](#) (title:string, body:string)

6.8.1 Member Function Documentation

6.8.1.1 createChannel()

```
async NotificationService::createChannel () [inline]
```

6.8.1.2 displayNotification()

```
async NotificationService::displayNotification (  
    title:string,  
    body:string) [inline]
```

6.8.1.3 requestPermission()

```
async NotificationService::requestPermission () [inline]
```

The documentation for this class was generated from the following file:

- [mobile/services/NotificationService.ts](#)

6.9 SensorRowProps Interface Reference

Propriétés de la ligne de capteurs.

6.9.1 Detailed Description

Propriétés de la ligne de capteurs.

The documentation for this interface was generated from the following file:

- [mobile/components/dashboard/SensorRow.tsx](#)

6.10 SetupPanelProps Interface Reference

Propriétés du panneau de configuration initiale.

6.10.1 Detailed Description

Propriétés du panneau de configuration initiale.

The documentation for this interface was generated from the following file:

- [mobile/components/SetupPanel.tsx](#)

6.11 StatusDisplayProps Interface Reference

Propriétés de l'affichage d'état.

6.11.1 Detailed Description

Propriétés de l'affichage d'état.

The documentation for this interface was generated from the following file:

- [mobile/components/dashboard/StatusDisplay.tsx](#)

6.12 TabHeaderProps Interface Reference

Propriétés de l'en-tête d'onglets.

6.12.1 Detailed Description

Propriétés de l'en-tête d'onglets.

The documentation for this interface was generated from the following file:

- [mobile/components/TabHeader.tsx](#)

6.13 WindowState Interface Reference

État de la fenêtre côté mobile.

6.13.1 Detailed Description

État de la fenêtre côté mobile.

The documentation for this interface was generated from the following file:

- [mobile/hooks/useWindowApi.ts](#)

Chapter 7

File Documentation

7.1 firmware/src/main.cpp File Reference

```
#include <main.hpp>
```

Classes

- class [ConfigCallbacks](#)
Traite les configurations WiFi et de localisation.

Functions

- WebServer [server](#) (3001)
- bool [checkSafety](#) ()
Vérifie l'état de sécurité du système.
- void [setVMC](#) (int angle)
Commande le servomoteur à un angle spécifique.
- void [handleStatus](#) ()
Gestionnaire HTTP GET /api/window/status.
- void [handleControl](#) ()
Gestionnaire HTTP POST /api/window/control.
- void [setup](#) ()
Initialisation de la connexion (WiFi, BLE, serveur).
- void [loop](#) ()
Boucle d'exécution principale.

Variables

- Servo [windowServo](#)
- Preferences [preferences](#)
- BLECharacteristic * [plpChar](#)
- String [wifi_ssid](#) = ""
- String [wifi_pass](#) = ""
- float [latitude](#) = 45.18
- float [longitude](#) = 5.72
- float [tempMax](#) = 30.0
- float [tempMin](#) = 18.0
- int [aqiMax](#) = 50
- bool [isOpen](#) = false
- bool [autoMode](#) = true
- int [targetAngle](#) = 0
- bool [safetyLockout](#) = false
- int [lastCurrentReading](#) = 0
- float [lastTemp](#) = 0.0
- int [lastAQI](#) = 0
- unsigned long [lastWeatherCheck](#) = 0
- int [currentAngle](#) = -1
- bool [shouldRestart](#) = false

7.1.1 Function Documentation

7.1.1.1 `checkSafety()`

```
bool checkSafety ()
```

Vérifie l'état de sécurité du système.

Lit le capteur de courant et vérifie s'il dépasse le seuil

Returns

true si sécurisé, false si surintensité détectée

7.1.1.2 `handleControl()`

```
void handleControl ()
```

Gestionnaire HTTP POST /api/window/control.

Reçoit les commandes pour contrôler la fenêtre

7.1.1.3 `handleStatus()`

```
void handleStatus ()
```

Gestionnaire HTTP GET /api/window/status.

Retourne l'état complet du système en JSON

7.1.1.4 loop()

```
void loop ()
```

Boucle d'exécution principale.

Gère les requêtes HTTP, la météo et les redémarrages

7.1.1.5 server()

```
WebServer server (  
    3001 )
```

7.1.1.6 setup()

```
void setup ()
```

Initialisation de la connexion (WiFi, BLE, serveur).

7.1.1.7 setVMC()

```
void setVMC (  
    int angle)
```

Commande le servomoteur à un angle spécifique.

Parameters

<i>angle</i>	Angle cible en degrés (0-90)
--------------	------------------------------

Vérifie la sécurité, contraint l'angle et met à jour l'état

7.1.2 Variable Documentation

7.1.2.1 aqiMax

```
int aqiMax = 50
```

7.1.2.2 autoMode

```
bool autoMode = true
```

7.1.2.3 currentAngle

```
int currentAngle = -1
```

7.1.2.4 isOpen

```
bool isOpen = false
```

7.1.2.5 lastAQI

```
int lastAQI = 0
```

7.1.2.6 lastCurrentReading

```
int lastCurrentReading = 0
```

7.1.2.7 lastTemp

```
float lastTemp = 0.0
```

7.1.2.8 lastWeatherCheck

```
unsigned long lastWeatherCheck = 0
```

7.1.2.9 latitude

```
float latitude = 45.18
```

7.1.2.10 longitude

```
float longitude = 5.72
```

7.1.2.11 plpChar

```
BLECharacteristic* pIpChar
```

7.1.2.12 preferences

```
Preferences preferences
```

7.1.2.13 safetyLockout

```
bool safetyLockout = false
```

7.1.2.14 shouldRestart

```
bool shouldRestart = false
```

7.1.2.15 targetAngle

```
int targetAngle = 0
```

7.1.2.16 tempMax

```
float tempMax = 30.0
```

7.1.2.17 tempMin

```
float tempMin = 18.0
```

7.1.2.18 wifi_pass

```
String wifi_pass = ""
```

7.1.2.19 wifi_ssid

```
String wifi_ssid = ""
```

7.1.2.20 windowServo

```
Servo windowServo
```

7.2 firmware/src/main.hpp File Reference

```
#include <Arduino.h>
#include <WiFi.h>
#include <HttpClient.h>
#include <WebServer.h>
#include <ArduinoJson.h>
#include <ESP32Servo.h>
#include <NimBLEDevice.h>
#include <Preferences.h>
```

Classes

- class [ConfigCallbacks](#)

Traite les configurations WiFi et de localisation.

Macros

- #define [SERVO_PIN](#) 13
- #define [CURRENT_SENSOR_PIN](#) 34
- #define [MAX_CURRENT_THRESHOLD](#) 2500
- #define [SERVICE_UUID](#) "4fafc201-1fb5-459e-8fcc-c5c9c331914b"
- #define [CHAR_CONFIG_UUID](#) "beb5483e-36e1-4688-b7f5-ea07361b26a8"
- #define [CHAR_IP_UUID](#) "12345678-1234-1234-1234-1234567890ab"

Functions

- bool [checkSafety](#) ()
Vérifie l'état de sécurité du système.
- void [setVMC](#) (int angle)
Commande le servomoteur à un angle spécifique.
- void [handleStatus](#) ()
Gestionnaire HTTP GET /api/window/status.
- void [handleControl](#) ()
Gestionnaire HTTP POST /api/window/control.
- void [setup](#) ()
Initialisation de la connexion (WiFi, BLE, serveur).
- void [loop](#) ()
Boucle d'exécution principale.

Variables

- Servo [windowServo](#)
- Preferences [preferences](#)
- WebServer [server](#)
- BLECharacteristic * [plpChar](#)
- String [wifi_ssid](#)
- String [wifi_pass](#)
- float [latitude](#)
- float [longitude](#)
- float [tempMax](#)
- float [tempMin](#)
- int [aqiMax](#)
- bool [isOpen](#)
- bool [autoMode](#)
- int [targetAngle](#)
- bool [safetyLockout](#)
- int [lastCurrentReading](#)
- int [currentAngle](#)
- float [lastTemp](#)
- int [lastAQI](#)
- unsigned long [lastWeatherCheck](#)
- bool [shouldRestart](#)

7.2.1 Macro Definition Documentation

7.2.1.1 CHAR_CONFIG_UUID

```
#define CHAR_CONFIG_UUID "beb5483e-36e1-4688-b7f5-ea07361b26a8"
```

7.2.1.2 CHAR_IP_UUID

```
#define CHAR_IP_UUID "12345678-1234-1234-1234-1234567890ab"
```

7.2.1.3 CURRENT_SENSOR_PIN

```
#define CURRENT_SENSOR_PIN 34
```

7.2.1.4 MAX_CURRENT_THRESHOLD

```
#define MAX_CURRENT_THRESHOLD 2500
```

7.2.1.5 SERVICE_UUID

```
#define SERVICE_UUID "4fafc201-1fb5-459e-8fcc-c5c9c331914b"
```

7.2.1.6 SERVO_PIN

```
#define SERVO_PIN 13
```

7.2.2 Function Documentation

7.2.2.1 checkSafety()

```
bool checkSafety ()
```

Vérifie l'état de sécurité du système.

Lit le capteur de courant et vérifie s'il dépasse le seuil

Returns

true si sécurisé, false si surintensité détectée

7.2.2.2 `handleControl()`

```
void handleControl ()
```

Gestionnaire HTTP POST `/api/window/control`.

Reçoit les commandes pour contrôler la fenêtre

7.2.2.3 `handleStatus()`

```
void handleStatus ()
```

Gestionnaire HTTP GET `/api/window/status`.

Retourne l'état complet du système en JSON

7.2.2.4 `loop()`

```
void loop ()
```

Boucle d'exécution principale.

Gère les requêtes HTTP, la météo et les redémarrages

7.2.2.5 `setup()`

```
void setup ()
```

Initialisation de la connexion (WiFi, BLE, serveur).

7.2.2.6 `setVMC()`

```
void setVMC (  
    int angle)
```

Commande le servomoteur à un angle spécifique.

Parameters

<i>angle</i>	Angle cible en degrés (0-90)
--------------	------------------------------

Vérifie la sécurité, contraint l'angle et met à jour l'état

7.2.3 Variable Documentation

7.2.3.1 `aqiMax`

```
int aqiMax [extern]
```

7.2.3.2 autoMode

```
bool autoMode [extern]
```

7.2.3.3 currentAngle

```
int currentAngle [extern]
```

7.2.3.4 isOpen

```
bool isOpen [extern]
```

7.2.3.5 lastAQI

```
int lastAQI [extern]
```

7.2.3.6 lastCurrentReading

```
int lastCurrentReading [extern]
```

7.2.3.7 lastTemp

```
float lastTemp [extern]
```

7.2.3.8 lastWeatherCheck

```
unsigned long lastWeatherCheck [extern]
```

7.2.3.9 latitude

```
float latitude [extern]
```

7.2.3.10 longitude

```
float longitude [extern]
```

7.2.3.11 plpChar

```
BLECharacteristic* pIpChar [extern]
```

7.2.3.12 preferences

Preferences preferences [extern]

7.2.3.13 safetyLockout

bool safetyLockout [extern]

7.2.3.14 server

WebServer server [extern]

7.2.3.15 shouldRestart

bool shouldRestart [extern]

7.2.3.16 targetAngle

int targetAngle [extern]

7.2.3.17 tempMax

float tempMax [extern]

7.2.3.18 tempMin

float tempMin [extern]

7.2.3.19 wifi_pass

String wifi_pass [extern]

7.2.3.20 wifi_ssid

String wifi_ssid [extern]

7.2.3.21 windowServo

Servo windowServo [extern]

7.3 main.hpp

[Go to the documentation of this file.](#)

```

00001 #ifndef MAIN_HPP
00002 #define MAIN_HPP
00003
00004 #include <Arduino.h>
00005 #include <WiFi.h>
00006 #include <HTTPClient.h>
00007 #include <WebServer.h>
00008 #include <ArduinoJson.h>
00009 #include <ESP32Servo.h>
00010 #include <NimBLEDevice.h>
00011 #include <Preferences.h>
00012
00013 #define SERVO_PIN 13
00014 #define CURRENT_SENSOR_PIN 34
00015 #define MAX_CURRENT_THRESHOLD 2500
00016 #define SERVICE_UUID "4fafc201-1fb5-459e-8fcc-c5c9c331914b"
00017 #define CHAR_CONFIG_UUID "beb5483e-36e1-4688-b7f5-ea07361b26a8"
00018 #define CHAR_IP_UUID "12345678-1234-1234-1234-1234567890ab"
00019
00020 extern Servo windowServo;
00021
00022
00023 extern Preferences preferences;
00024
00025
00026 extern WebServer server;
00027
00028
00029 extern BLECharacteristic *pIpChar;
00030
00031
00032 extern String wifi_ssid;
00033 extern String wifi_pass;
00034
00035 extern float latitude;
00036 extern float longitude;
00037 extern float tempMax;
00038 extern float tempMin;
00039 extern int aqiMax;
00040
00041 extern bool isOpen;
00042 extern bool autoMode;
00043 extern int targetAngle;
00044
00045 extern bool safetyLockout;
00046
00047 extern int lastCurrentReading;
00048 extern int currentAngle;
00049
00050 extern float lastTemp;
00051 extern int lastAQI;
00052 extern unsigned long lastWeatherCheck;
00053
00054 extern bool shouldRestart;
00055
00061 bool checkSafety();
00062
00068 void setVMC(int angle);
00069
00074 void handleStatus();
00075
00080 void handleControl();
00081
00085 void setup();
00090 void loop();
00091
00093
00098 class ConfigCallbacks : public BLECharacteristicCallbacks {
00103     void onWrite(BLECharacteristic *pCharacteristic);
00104 };
00105
00106 #endif // MAIN_HPP

```

7.4 mobile/.eslintrc.js File Reference

Variables

- module [exports](#)

7.4.1 Variable Documentation

7.4.1.1 exports

module exports

Initial value:

```
= {
  root: true,
  extends: '@react-native',
}
```

7.5 mobile/.prettierrc.js File Reference

Variables

- module [exports](#)

7.5.1 Variable Documentation

7.5.1.1 exports

module exports

Initial value:

```
= {
  arrowParens: 'avoid',
  singleQuote: true,
  trailingComma: 'all',
}
```

7.6 mobile/App.tsx File Reference

Composant principal de l'application mobile Smart VMC.

Variables

- [import React](#)
- import [import](#) { SafeAreaView, ScrollView, StatusBar, StyleSheet, Alert, ActivityIndicator, View, Text } from 'react-native'
- [import](#) AsyncStorage from [react](#) native async storage [async storage](#)
- [import](#) TabHeader from components [TabHeader](#)
- [import](#) SetupPanel from components [SetupPanel](#)
- [import](#) DashboardPanel from components [DashboardPanel](#)
- [import](#) ConfigurationPanel from components [ConfigurationPanel](#)
- function App() export default [App](#)

@function App

7.6.1 Detailed Description

Composant principal de l'application mobile Smart VMC.

Gère la navigation entre les onglets (Setup, Dashboard, Configuration) et la communication avec l'ESP32 via BLE et API HTTP

7.6.2 Variable Documentation

7.6.2.1 App

```
function App () export default App ()
```

@function App

Composant racine de l'application Smart VMC

Gère :

- La navigation entre 3 onglets (Setup, Dashboard, Configuration)
- La connexion BLE à l'ESP32
- La communication avec l'API REST
- Le stockage persistant de l'adresse IP du serveur

Returns

{React.JSX.Element} Interface utilisateur complète

7.6.2.2 ConfigurationPanel

```
export default ConfigurationPanel
```

7.6.2.3 DashboardPanel

```
export default DashboardPanel
```

7.6.2.4 import

```
import { SafeAreaView, ScrollView, StatusBar, StyleSheet, Alert, ActivityIndicator, View, Text  
} from 'react-native'
```

7.6.2.5 React

```
import React
```

7.6.2.6 SetupPanel

```
export default SetupPanel
```

7.6.2.7 storage

```
import AsyncStorage from react native async storage async storage
```

7.6.2.8 TabHeader

```
export default TabHeader
```

7.7 mobile/babel.config.js File Reference

Variables

- module `exports`

7.7.1 Variable Documentation

7.7.1.1 exports

```
module exports
```

Initial value:

```
= {
  presets: ['module:@react-native/babel-preset'],
}
```

7.8 mobile/components/ConfigurationPanel.tsx File Reference

Panneau de configuration des seuils de température et pollution.

Variables

- `import React`
- `import import { View, Text, StyleSheet, Alert } from 'react-native'`
- `onUpdateThresholds __pad0__`
- `onUpdateThresholds tMin`
- `onUpdateThresholds aqiMax`
- `isLoading __pad1__`
- `const ConfigurationPanel`
 - @function ConfigurationPanel*
- `const onUpdateThresholds`
- `const isLoading`
- `const styles`

7.8.1 Detailed Description

Panneau de configuration des seuils de température et pollution.

7.8.2 Variable Documentation

7.8.2.1 __pad0__

```
onUpdateThresholds __pad0__
```

7.8.2.2 __pad1__

```
isLoading __pad1__
```

7.8.2.3 aqiMax

```
onUpdateThresholds aqiMax
```

7.8.2.4 ConfigurationPanel

```
export default ConfigurationPanel
```

```
@function ConfigurationPanel
```

Formulaire de configuration des seuils (température, AQI) pour le mode Auto

Parameters

<code>{ConfigurationPanelProps}</code>	props - Propriétés du composant
--	---------------------------------

Returns

`{React.JSX.Element}` Carte de configuration

7.8.2.5 import

```
import { View, Text, StyleSheet, Alert } from 'react-native'
```

7.8.2.6 isLoading

```
const isLoading
```

7.8.2.7 onUpdateThresholds

```
const onUpdateThresholds
```

7.8.2.8 React

```
import React
```

7.8.2.9 styles

```
const styles
```

7.8.2.10 tMin

```
onUpdateThresholds tMin
```

7.9 mobile/components/dashboard/ControlPanel.tsx File Reference

Panneau de contrôle de la VMC (mode auto, commandes manuelles, slider).

Variables

- `import React` from `react`
- `import { View, Text, Switch, StyleSheet }` from `'react-native'`
- `import Slider` from `react native community slider`
- `isLocked __pad0__`
- `targetAngle __pad1__`
- `onToggleAuto __pad2__`
- `onCommand __pad3__`
- `onAngleChange __pad4__`
- `onResetSafety __pad5__`
- `export const ControlPanel`
 - @function ControlPanel*
- `export const isLocked`
- `export const targetAngle`
- `export const onToggleAuto`
- `export const onCommand`
- `export const onAngleChange`
- `export const onResetSafety`
- `const styles`

7.9.1 Detailed Description

Panneau de contrôle de la VMC (mode auto, commandes manuelles, slider).

7.9.2 Variable Documentation

7.9.2.1 __pad0__

`isLocked` __pad0__

7.9.2.2 __pad1__

`targetAngle` __pad1__

7.9.2.3 __pad2__

`onToggleAuto` __pad2__

7.9.2.4 __pad3__

`onCommand` __pad3__

7.9.2.5 __pad4__

`onAngleChange` __pad4__

7.9.2.6 __pad5__

`onResetSafety` __pad5__

7.9.2.7 ControlPanel

```
export const ControlPanel
```

```
@function ControlPanel
```

Panneau de contrôle principal de la VMC

Parameters

<code>{ControlPanelProps}</code>	props - Propriétés du composant
----------------------------------	---------------------------------

Returns

`{React.JSX.Element}` Panneau de contrôle

Affiche deux modes :

- Mode verrouillé : Bouton de réarmement sécurité
- Mode normal : Switch auto/manuel, boutons ouvrir/fermer, slider d'angle

7.9.2.8 import

```
import { View, Text, Switch, StyleSheet } from 'react-native'
```

7.9.2.9 isLocked

```
export const isLocked
```

7.9.2.10 onAngleChange

```
export const onAngleChange
```

7.9.2.11 onCommand

```
export const onCommand
```

7.9.2.12 onResetSafety

```
export const onResetSafety
```

7.9.2.13 onToggleAuto

```
export const onToggleAuto
```

7.9.2.14 react

```
import React from react
```

7.9.2.15 slider

```
import Slider from react native community slider
```

7.9.2.16 styles

```
const styles
```

7.9.2.17 targetAngle

```
export const targetAngle
```

7.10 mobile/components/dashboard/SensorRow.tsx File Reference

Ligne d'affichage des valeurs de capteurs (température et AQI).

Variables

- `import React` from `react`
- `import { View, Text, StyleSheet }` from `'react-native'`
- `aqi __pad0__`
- `export const SensorRow`
`@function SensorRow`
- `export const aqi`
- `const styles`

7.10.1 Detailed Description

Ligne d'affichage des valeurs de capteurs (température et AQI).

7.10.2 Variable Documentation

7.10.2.1 `__pad0__`

`aqi __pad0__`

7.10.2.2 `aqi`

`export const aqi`

Initial value:

```
=> {
  return (
    <View style={styles.row}>
      <Text style={styles.value}> {temp !== undefined && temp !== null ? temp.toFixed(1) : '--'}°C</Text>
      <Text style={styles.value}> AQI {aqi ?? '--'}</Text>
    </View>
  );
}
```

7.10.2.3 `import`

`import { View, Text, StyleSheet } from 'react-native'`

7.10.2.4 `react`

`import React` from `react`

7.10.2.5 `SensorRow`

`export const SensorRow`

`@function SensorRow`

Affiche la température et l'AQI sous forme de badges

Parameters

<code>{SensorRowProps}</code>	props - Propriétés du composant
-------------------------------	---------------------------------

Returns

`{React.JSX.Element}` Ligne de capteurs

7.10.2.6 styles

`const styles`

Initial value:

```
= StyleSheet.create({
  row: {
    flexDirection: 'row',
    justifyContent: 'center',
    gap: 20,
    marginBottom: 25,
  },
  value: {
    fontSize: 18,
    fontWeight: '600',
    color: '#374151',
  },
})
```

7.11 mobile/components/dashboard/StatusDisplay.tsx File Reference

Affichage de l'état global de la VMC (ouverte/fermée/verrouillée).

Variables

- `import React` from `react`
- `import { View, Text, StyleSheet }` from `'react-native'`
- `isLocked __pad0__`
- export `const StatusDisplay`
 @function StatusDisplay
- export `const isLocked`
- `const styles`

7.11.1 Detailed Description

Affichage de l'état global de la VMC (ouverte/fermée/verrouillée).

7.11.2 Variable Documentation

7.11.2.1 __pad0__

`isLocked __pad0__`

7.11.2.2 import

```
import { View, Text, StyleSheet } from 'react-native'
```

7.11.2.3 isLocked

```
export const isLocked
```

Initial value:

```
=> {  
  
  if (isLocked) {  
    return (  
      <View style={styles.container}>  
        <Text style={styles.emoji}></Text>  
        <Text style={[styles.text, { color: '#DC2626' }]}>  
          SÉCURITÉ ACTIVE  
        </Text>  
        <Text style={styles.subtext}>Moteur bloqué</Text>  
      </View>  
    );  
  }  
  
  return (  
    <View style={styles.container}>  
      <Text style={styles.emoji}>  
        {isOpen ? '' : ''}  
      </Text>  
      <Text style={ [  
        styles.text,  
        { color: isOpen ? '#007A5E' : '#EF4444' }  
      ]}>  
        {isOpen ? 'OUVERTE' : 'FERMÉE'}  
      </Text>  
    </View>  
  );  
}
```

7.11.2.4 react

```
import React from react
```

7.11.2.5 StatusDisplay

```
export const StatusDisplay
```

```
@function StatusDisplay
```

Affiche l'état visuel de la VMC avec emoji et texte coloré

Parameters

<code>{StatusDisplayProps}</code>	props - Propriétés du composant
-----------------------------------	---------------------------------

Returns

`{React.JSX.Element}` Vue d'état

7.11.2.6 styles

`const styles`

Initial value:

```
= StyleSheet.create({
  container: {
    alignItems: 'center',
    marginBottom: 20,
  },
  emoji: {
    fontSize: 80,
    marginBottom: 10,
  },
  text: {
    fontSize: 28,
    fontWeight: '800',
    letterSpacing: 0.5,
    textAlign: 'center',
  },
  subtext: {
    fontSize: 16,
    color: '#666',
    marginTop: 5,
  }
})
```

7.12 mobile/components/DashboardPanel.tsx File Reference

Tableau de bord affichant l'état, les capteurs et les commandes de la VMC.

Variables

- `import React`
- `import import { StyleSheet, View } from 'react-native'`
- `import NotificationService from services NotificationService`
- `onToggleAuto __pad0__`
- `onCommand __pad1__`
- `onAngleChange __pad2__`
- `onResetSafety __pad3__`
- `const DashboardPanel`
@function DashboardPanel
- `const onToggleAuto`
- `const onCommand`
- `const onAngleChange`
- `const onResetSafety`
- `const styles`

7.12.1 Detailed Description

Tableau de bord affichant l'état, les capteurs et les commandes de la VMC.

7.12.2 Variable Documentation

7.12.2.1 __pad0__

`onToggleAuto __pad0__`

7.12.2.2 __pad1__

`onCommand __pad1__`

7.12.2.3 __pad2__

`onAngleChange __pad2__`

7.12.2.4 __pad3__

`onResetSafety __pad3__`

7.12.2.5 DashboardPanel

```
export default DashboardPanel
```

```
@function DashboardPanel
```

Affiche l'état, les capteurs et les commandes de la VMC

Parameters

<code>{DashboardPanelProps}</code>	props - Propriétés du composant
------------------------------------	---------------------------------

Returns

`{React.JSX.Element}` Vue tableau de bord

7.12.2.6 import

```
import { StyleSheet, View } from 'react-native'
```

7.12.2.7 NotificationService

```
import NotificationService from services NotificationService
```

7.12.2.8 onAngleChange

```
const onAngleChange
```

7.12.2.9 onCommand

```
const onCommand
```

7.12.2.10 onResetSafety

```
const onResetSafety
```

7.12.2.11 onToggleAuto

```
const onToggleAuto
```

7.12.2.12 React

```
import React
```

7.12.2.13 styles

```
const styles
```

Initial value:

```
= StyleSheet.create({
  container: {
    alignItems: 'center',
  },
})
```

7.13 mobile/components/SetupPanel.tsx File Reference

Panneau de configuration initiale avec saisie WiFi et localisation.

Variables

- `import React`
- `import import { View, Text, StyleSheet, Platform, Alert } from 'react-native'`
- `import Geolocation from react native geolocation service`
- `isScanning __pad0__`
- `onConnect __pad1__`
- `onConnect pass`
- `onConnect lat`
- `onConnect lon`
- `onConnect ip`
- `savedIp __pad2__`
- `const SetupPanel`
 - *@function SetupPanel*
- `const isScanning`
- `const onConnect`
- `const savedIp`
- `const styles`

7.13.1 Detailed Description

Panneau de configuration initiale avec saisie WiFi et localisation.

7.13.2 Variable Documentation

7.13.2.1 __pad0__

`isScanning` __pad0__

7.13.2.2 __pad1__

`onConnect` __pad1__

7.13.2.3 __pad2__

`savedIp` __pad2__

7.13.2.4 import

```
import { View, Text, StyleSheet, Platform, Alert } from 'react-native'
```

7.13.2.5 ip

`onConnect` ip

7.13.2.6 isScanning

`const` isScanning

7.13.2.7 lat

`onConnect` lat

7.13.2.8 lon

`onConnect` lon

7.13.2.9 onConnect

`const` onConnect

7.13.2.10 pass

```
onConnect pass
```

7.13.2.11 React

```
import React
```

7.13.2.12 savedIp

```
const savedIp
```

7.13.2.13 service

```
import Geolocation from react native geolocation service
```

7.13.2.14 SetupPanel

```
export default SetupPanel
```

```
@function SetupPanel
```

Formulaire d'envoi des identifiants WiFi + localisation via BLE

Parameters

<code>{SetupPanelProps}</code>	props - Propriétés du composant
--------------------------------	---------------------------------

Returns

`{React.JSX.Element}` Carte de configuration initiale

7.13.2.15 styles

```
const styles
```

Initial value:

```
= StyleSheet.create({
  sectionTitle: { fontSize: 20, fontWeight: '700', marginBottom: 15, color: '#007A5E' },
  statusText: { marginBottom: 15, textAlign: 'center', color: '#666', fontSize: 14 },
})
```

7.14 mobile/components/TabHeader.tsx File Reference

Barre d'onglets principale.

Variables

- `import React` from `react`
- `import { View, Text, TouchableOpacity, StyleSheet }` from `'react-native'`
- `onTabChange __pad0__`
- `const TabHeader`
@function TabHeader
- `const onTabChange`
- `const styles`

7.14.1 Detailed Description

Barre d'onglets principale.

7.14.2 Variable Documentation

7.14.2.1 `__pad0__`

`onTabChange __pad0__`

7.14.2.2 `import`

`import { View, Text, TouchableOpacity, StyleSheet }` from `'react-native'`

7.14.2.3 `onTabChange`

`const onTabChange`

Initial value:

```
=> {
  return (
    <View style={styles.header}>
      <Text style={styles.headerTitle}> Smart VMC</Text>
      <View style={styles.tabs}>
        <TouchableOpacity
          onPress={() => onTabChange('SETUP')}
          style={[styles.tab, activeTab === 'SETUP' && styles.activeTab]}
        >
          <Text style={styles.tabText}> Wifi Setup</Text>
        </TouchableOpacity>
        <TouchableOpacity
          onPress={() => onTabChange('DASHBOARD')}
          style={[styles.tab, activeTab === 'DASHBOARD' && styles.activeTab]}
        >
          <Text style={styles.tabText}> Contrôle</Text>
        </TouchableOpacity>
        <TouchableOpacity
          onPress={() => onTabChange('CONFIGURATION')}
          style={[styles.tab, activeTab === 'CONFIGURATION' && styles.activeTab]}
        >
          <Text style={styles.tabText}> Réglages</Text>
        </TouchableOpacity>
      </View>
    </View>
  );
}
```

7.14.2.4 react

```
import React from react
```

7.14.2.5 styles

```
const styles
```

7.14.2.6 TabHeader

```
export default TabHeader
```

```
@function TabHeader
```

Barre d'onglets principale

Parameters

<code>{TabHeaderProps}</code>	props - Propriétés du composant
-------------------------------	---------------------------------

Returns

`{React.JSX.Element}` En-tête avec trois onglets

7.15 mobile/components/ui/AppButton.tsx File Reference

Bouton applicatif stylé (primaire, danger, outline) avec état de chargement.

Variables

- `import React` from `react`
- `import { TouchableOpacity, Text, StyleSheet, TouchableOpacityProps, ActivityIndicator }` from `'react-native'`
- interface `AppButtonProps` extends `TouchableOpacityProps`
- `variant __pad0__`
- `isLoading __pad1__`
- export `const AppButton`
 - `@function AppButton`
- export `const variant` = `'primary'`
- export `const isLoading` = `false`
- export `const style`
- export `const disabled`
- export `const props`
- `const styles`

7.15.1 Detailed Description

Bouton applicatif stylé (primaire, danger, outline) avec état de chargement.

7.15.2 Variable Documentation

7.15.2.1 __pad0__

```
variant __pad0__
```

7.15.2.2 __pad1__

```
isLoading __pad1__
```

7.15.2.3 AppButton

```
export const AppButton
```

```
@function AppButton
```

Bouton réutilisable avec variantes et état de chargement

Parameters

<code>{AppButtonProps}</code>	props - Propriétés du composant
-------------------------------	---------------------------------

Returns

`{React.JSX.Element}` Bouton stylé

7.15.2.4 disabled

```
export const disabled
```

7.15.2.5 import

```
import { TouchableOpacity, Text, StyleSheet, TouchableOpacityProps, ActivityIndicator } from  
'react-native'
```

7.15.2.6 isLoading

```
export const isLoading = false
```

7.15.2.7 props

```
export const props
```

7.15.2.8 react

```
import React from react
```

7.15.2.9 style

```
export const style
```

7.15.2.10 styles

```
const styles
```

Initial value:

```
= StyleSheet.create({
  base: {
    paddingVertical: 14,
    borderRadius: 12,
    alignItems: 'center',
    justifyContent: 'center',
    shadowColor: '#000',
    shadowOffset: { width: 0, height: 2 },
    shadowOpacity: 0.1,
    shadowRadius: 3,
    elevation: 2,
  },
  text: {
    fontWeight: '700',
    fontSize: 16,
    letterSpacing: 0.5,
  },
  disabled: {
    opacity: 0.6,
  }
})
```

7.15.2.11 TouchableOpacityProps

```
interface AppButtonProps extends TouchableOpacityProps
```

Initial value:

```
{
  title: string
```

7.15.2.12 variant

```
export const variant = 'primary'
```

7.16 mobile/components/ui/AppInput.tsx File Reference

Champ de saisie stylé avec placeholder grisé

Variables

- `import React` from `react`
- `import { TextInput, StyleSheet, TextInputProps }` from `'react-native'`
- interface `AppInputProps` extends `TextInputProps`
- `const styles`

7.16.1 Detailed Description

Champ de saisie stylé avec placeholder grisé

7.16.2 Variable Documentation

7.16.2.1 import

```
import { TextInput, StyleSheet, TextInputProps } from 'react-native'
```

7.16.2.2 react

```
import React from react
```

7.16.2.3 styles

```
const styles
```

Initial value:

```
= StyleSheet.create({
  input: {
    borderWidth: 1,
    borderColor: '#E0E0E0',
    borderRadius: 12,
    paddingVertical: 14,
    paddingHorizontal: 16,
    marginBottom: 15,
    fontSize: 16,
    backgroundColor: '#F9FAFB',
    color: '#1F2937',
  },
})
```

7.16.2.4 TextInputProps

```
interface AppInputProps extends TextInputProps
```

Initial value:

```
{}
```

```
export const AppInput: React.FC<AppInputProps> = ({ style, ...props }) => {
  return (
    <TextInput
      style={[styles.input, style]}
      placeholderTextColor="#9CA3AF"
      {...props}
    />
  );
}
```

7.17 mobile/components/ui/Card.tsx File Reference

Conteneur carte avec ombre et arrondis.

Variables

- `import React` from `react`
- `import { View, StyleSheet, ViewStyle, ViewProps }` from `'react-native'`
- interface `CardProps` extends `ViewProps`
- `style __pad0__`
- `export const Card`
- `@function Card`
- `export const style`
- `export const props`
- `const styles`

7.17.1 Detailed Description

Conteneur carte avec ombre et arrondis.

7.17.2 Variable Documentation

7.17.2.1 `__pad0__`

`style __pad0__`

7.17.2.2 `Card`

`export const Card`

`@function Card`

Conteneur stylé type carte

Parameters

<code>{CardProps}</code>	props - Propriétés du composant
--------------------------	---------------------------------

Returns

`{React.JSX.Element}` Carte

7.17.2.3 `import`

`import { View, StyleSheet, ViewStyle, ViewProps }` from `'react-native'`

7.17.2.4 props

```
export const props
```

Initial value:

```
=> {  
  return <View style={[styles.card, style]} {...props}>{children}</View>;  
}
```

7.17.2.5 react

```
import React from react
```

7.17.2.6 style

```
export const style
```

7.17.2.7 styles

```
const styles
```

Initial value:

```
= StyleSheet.create({  
  card: {  
    backgroundColor: 'white',  
    padding: 24,  
    borderRadius: 16,  
    marginBottom: 20,  
    shadowColor: '#000',  
    shadowOffset: { width: 0, height: 2 },  
    shadowOpacity: 0.1,  
    shadowRadius: 4,  
    elevation: 4,  
  },  
})
```

7.17.2.8 ViewProps

```
interface CardProps extends ViewProps
```

Initial value:

```
{  
  children: React.ReactNode
```

7.18 mobile/hooks/useBLE.ts File Reference

Hook React pour la communication Bluetooth avec l'ESP32.

Variables

- `import { useState, useEffect } from 'react'`
- `const SERVICE_UUID = "4fafc201-1fb5-459e-8fcc-c5c9c331914b"`
@constant {string} SERVICE_UUID
- `const CHAR_UUID = "beb5483e-36e1-4688-b7f5-ea07361b26a8"`
@constant {string} CHAR_UUID
- `const CHAR_IP_UUID = "12345678-1234-1234-1234-1234567890ab"`
@constant {string} CHAR_IP_UUID
- `const bleManager = new BleManager()`
@constant {BleManager} bleManager
- `export const useBLE`
@function useBLE

7.18.1 Detailed Description

Hook React pour la communication Bluetooth avec l'ESP32.

Gère le scan BLE, la connexion à l'ESP32, et l'échange de données

7.18.2 Variable Documentation

7.18.2.1 bleManager

```
const bleManager = new BleManager()
```

@constant {BleManager} bleManager

Instance globale du gestionnaire BLE

7.18.2.2 CHAR_IP_UUID

```
const CHAR_IP_UUID = "12345678-1234-1234-1234-1234567890ab"
```

@constant {string} CHAR_IP_UUID

UUID de la caractéristique BLE pour lire l'adresse IP de l'appareil

7.18.2.3 CHAR_UUID

```
const CHAR_UUID = "beb5483e-36e1-4688-b7f5-ea07361b26a8"
```

@constant {string} CHAR_UUID

UUID de la caractéristique BLE pour la configuration WiFi et localisation

7.18.2.4 import

```
import { useState, useEffect } from 'react'
```

7.18.2.5 SERVICE_UUID

```
const SERVICE_UUID = "4fafc201-1fb5-459e-8fcc-c5c9c331914b"
```

@constant {string} SERVICE_UUID

UUID du service BLE principal

7.18.2.6 useBLE

```
export const useBLE
```

@function useBLE

Hook pour gérer les opérations Bluetooth

Fournit :

- Le statut de la connexion BLE
- L'état du scan
- Une fonction pour configurer l'ESP32 via BLE
- Une fonction pour récupérer l'IP de l'ESP32 via BLE

Returns

{Object} Objet contenant les états et fonctions BLE
{string} bleStatus - Message de statut de la connexion BLE
{boolean} isScanning - Indique si un scan est en cours
{Function} scanAndConfigure - Configure l'ESP32 avec WiFi/GPS
{Function} scanAndGetIp - Récupère l'adresse IP de l'ESP32

7.19 mobile/hooks/useWindowApi.ts File Reference

Hook React pour communiquer avec l'API smart-vmc.

Variables

- `import { useState, useCallback } from 'react'`
- `export interface WindowState`
- `temp __pad0__`
- `aqi __pad1__`
- `targetAngle __pad2__`
- `autoMode __pad3__`
- `safetyLockout __pad4__`
- `currentLoad __pad5__`
- `tMax __pad6__`
- `tMin __pad7__`
- `aqiMax __pad8__`
- `export const useWindowApi`
 @function useWindowApi
- `const sendAngle`
 @function sendAngle
- `const toggleAutoMode`
 @function toggleAutoMode
- `const resetSafety`
 @function resetSafety
- `const updateThresholds`
 @function updateThresholds
- `return`

7.19.1 Detailed Description

Hook React pour communiquer avec l'API smart-vmc.

Fournit les appels REST pour lire l'état et piloter la VMC (ouverture, mode auto, seuils, sécurité).

7.19.2 Variable Documentation

7.19.2.1 __pad0__

`temp __pad0__`

7.19.2.2 __pad1__

`aqi __pad1__`

7.19.2.3 __pad2__

`targetAngle __pad2__`

7.19.2.4 __pad3__

`autoMode` __pad3__

7.19.2.5 __pad4__

`safetyLockout` __pad4__

7.19.2.6 __pad5__

`currentLoad` __pad5__

7.19.2.7 __pad6__

`tMax` __pad6__

7.19.2.8 __pad7__

`tMin` __pad7__

7.19.2.9 __pad8__

`aqiMax` __pad8__

7.19.2.10 import

```
import { useState, useCallback } from 'react'
```

7.19.2.11 resetSafety

```
const resetSafety
```

Initial value:

```
= async () => {  
  try {  
    const response = await fetch(`http:  
      method: 'POST',  
      body: JSON.stringify({ resetSafety: true })  
    });  
    const json = await response.json();  
    setWindowState(json.state || json);  
    Alert.alert("Info", "Système réarmé. Vérifiez qu'il n'y a plus d'obstruction.");  
  } catch (e) {  
    Alert.alert("Erreur", "Impossible de réarmer");  
  }  
}
```

@function resetSafety

Réarme le verrouillage de sécurité

7.19.2.12 return

return

Initial value:

```
{
  windowState,
  fetchStatus,
  sendCommand,
  sendAngle,
  toggleAutoMode,
  resetSafety,
  updateThresholds,
  isLoading
}
```

7.19.2.13 sendAngle

const sendAngle

Initial value:

```
= async (angle: number) => {
  try {
    const response = await fetch(`http:
      method: 'POST',
      body: JSON.stringify({ angle })
    });
    const json = await response.json();
    setWindowState(json.state || json);
  } catch (e) {
    console.log(e);
  }
}
```

@function sendAngle

Positionne la fenêtre à un angle précis

Parameters

<i>{number}</i>	angle - Angle cible (0-90)
-----------------	----------------------------

7.19.2.14 toggleAutoMode

const toggleAutoMode

Initial value:

```
= async (value: boolean) => {
  try {
    const response = await fetch(`http:
      method: 'POST',
      body: JSON.stringify({ autoMode: value })
    });
    const json = await response.json();
    setWindowState(json.state || json);
  } catch (e) {
    Alert.alert("Erreur", "Impossible de changer le mode");
  }
}
```

@function toggleAutoMode

Active/désactive le mode automatique

Parameters

<code>{boolean}</code>	value - true pour activer
------------------------	---------------------------

7.19.2.15 updateThresholds

`const` updateThresholds

Initial value:

```
= async (tMax: number, tMin: number, aqiMax: number) => {
  setIsLoading(true);
  try {
    const response = await fetch(`http:
      method: 'POST',
      body: JSON.stringify({ tMax, tMin, aqiMax })
    });
    const json = await response.json();
    setWindowState(json.state || json);
    Alert.alert("Succès", "Paramètres sauvegardés !");
  } catch (e) {
    Alert.alert("Erreur", "Echec sauvegarde");
  } finally {
    setIsLoading(false);
  }
}
```

@function updateThresholds

Met à jour les seuils température/AQI

Parameters

<code>{number}</code>	tMax - Seuil max
<code>{number}</code>	tMin - Seuil min
<code>{number}</code>	aqiMax - Seuil AQI max

7.19.2.16 useWindowApi

`export const` useWindowApi

@function useWindowApi

Hook pour appeler l'API HTTP de l'ESP32

Parameters

<code>{string}</code>	serverIp - Adresse IP de l'ESP32 (ex: "192.168.1.50")
-----------------------	---

Returns

`{object}` API du hook

`{WindowState|null}` return.windowState - Dernier état connu

`() => Promise<void>` return.fetchStatus - Récupère l'état courant

`{(action:'open'|'close') => Promise<void>}` return.sendCommand - Ouvre/ferme

`{(angle:number) => Promise<void>}` return.sendAngle - Définit un angle précis

`{(value:boolean) => Promise<void>}` return.toggleAutoMode - Active/désactive le mode auto

`() => Promise<void>` return.resetSafety - Réarme la sécurité

`{(tMax:number,tMin:number,aqiMax:number)=>Promise<void>}` return.updateThresholds - Met à jour les seuils

`{boolean}` return.isLoading - Indique si une requête de seuils est en cours

7.19.2.17 WindowState

```
export interface WindowState
```

Initial value:

```
{  
  isOpen: boolean
```

7.20 mobile/index.js File Reference

Variables

- `import` { AppRegistry } from 'react-native'
@format
- `import` App from App

7.20.1 Variable Documentation

7.20.1.1 App

```
import App from App
```

7.20.1.2 import

```
import { AppRegistry } from 'react-native'
```

@format

7.21 mobile/jest.config.js File Reference

Variables

- module `exports`

7.21.1 Variable Documentation

7.21.1.1 exports

```
module exports
```

Initial value:

```
= {  
  preset: 'react-native',  
}
```

7.22 mobile/metro.config.js File Reference

Variables

- `const { getDefaultConfig, mergeConfig } = require('@react-native/metro-config')`
- `const config = {}`
Metro configuration <https://reactnative.dev/docs/metro>.
- `module exports = mergeConfig(getDefaultConfig(__dirname), config)`

7.22.1 Variable Documentation

7.22.1.1 config

```
const config = {}
```

Metro configuration <https://reactnative.dev/docs/metro>.

```
@type {import('@react-native/metro-config').MetroConfig}
```

7.22.1.2 const

```
const { getDefaultConfig, mergeConfig } = require('@react-native/metro-config')
```

7.22.1.3 exports

```
module exports = mergeConfig(getDefaultConfig(__dirname), config)
```

7.23 mobile/services/NotificationService.ts File Reference

Classes

- class [NotificationService](#)

Variables

- `import notifee`

7.23.1 Variable Documentation

7.23.1.1 notifee

```
import notifee
```

