Overall structure

HA Addon
Tesla-local-control-addon
https://github.com/tesla-local-control/tesla-local-control-addon

[submodule "tesla_ble_mqtt/app"]
    path = tesla_ble_mqtt/app
    url = https://github.com/tesla-local-control/tesla_ble_mqtt_core
    branch = main

.gitmodules
tesla_ble_mqtt/
    app/ → *Added via .gitmodules*
    translations/
    CHANGELOG.md
    DOCS.md
    Dockerfile
    README.md
    config.yaml
    icon.png
    logo.png
LICENSE
repository.yaml → used to advertise contents of added addon repo in HA
README.md → associated Readme

Per HA addon in the repo

Specific to HA addon repo

Core
tesla_ble_mqtt_core
https://github.com/tesla-local-control/tesla_ble_mqtt_core

run.sh
listen_to_mqtt.sh
subroutines.sh
libcolor.sh

Standalone docker
Tesla_ble_mqtt_docker
https://github.com/tesla-local-control/tesla_ble_mqtt_docker

[submodule "app"]
    path = app
    url = https://github.com/tesla-local-control/tesla_ble_mqtt_core
    branch = main

.gitmodules
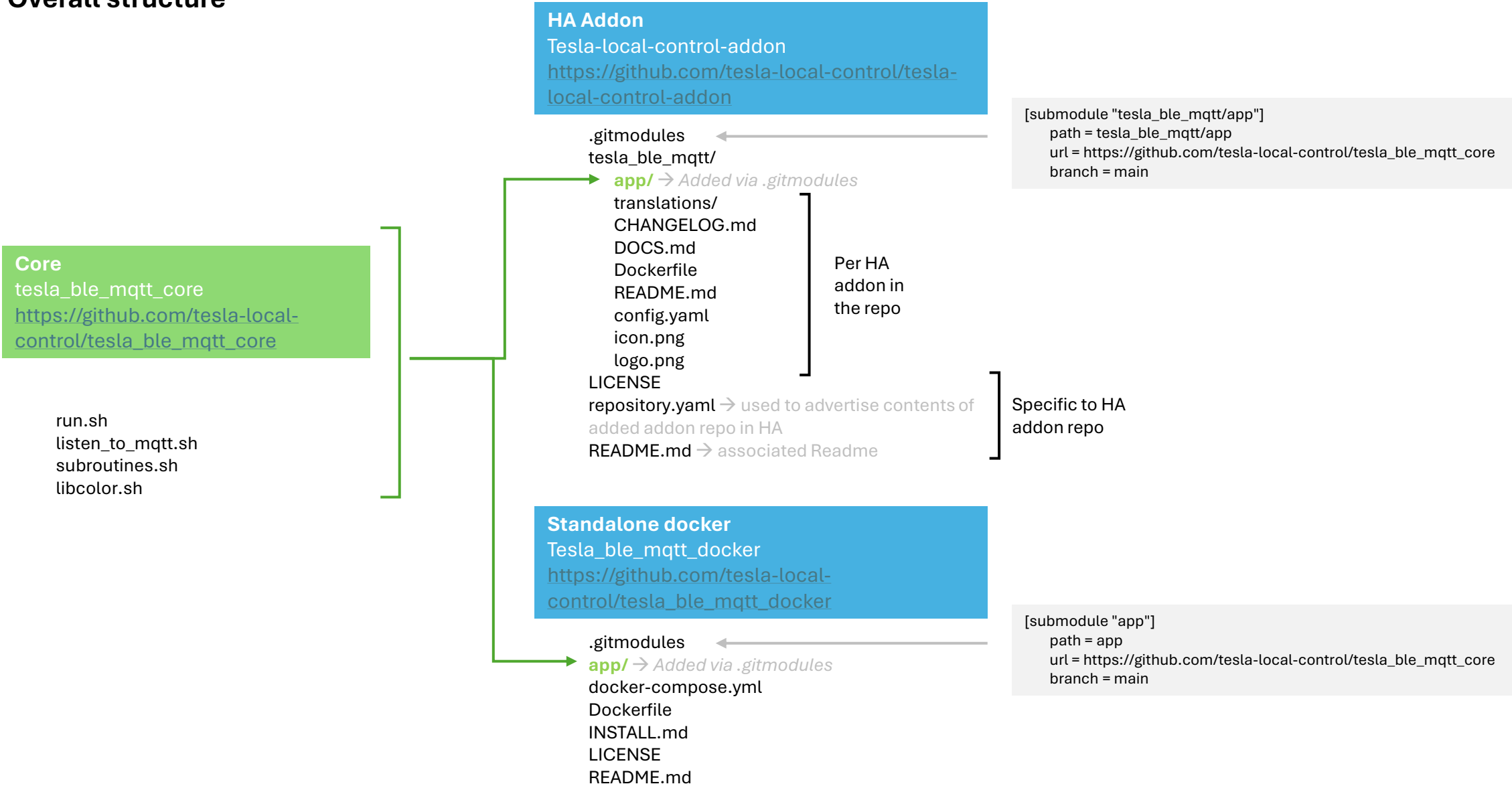**app/** → *Added via .gitmodules*
docker-compose.yml
Dockerfile
INSTALL.md
LICENSE
README.md

How to contribute

All the main code is in _core repo: all .sh files shall be modified here.
Interface with docker is in tesla_ble_mqtt_docker
Interface with HA is in tesla-local-control-addon

Core comits must be done in the _core repo

For minor changes (code cleaning, logging, …): no need to check compatibility with docker & addon

For more impacting changes, the compatibility shall be ensured (eg. add new core functionalities requiring additionnal packages, new environment variables, adding key functionalities, …)

To prepare a PR:
1. Branch _core + at least one of the two « applications » repo
2. Develop what you need
3. To update _core files in the application: run from command line:
    1. For HA Addon: cd tesla_ble_mqtt/app ; git pull https://github.com/tesla-local-control/tesla_ble_mqtt_core [your _core branch name]
    2. For docker: cd app ; git pull https://github.com/tesla-local-control/tesla_ble_mqtt_core [your _core branch name]
    3. Then commit the change of version
4. Test your application (addon or docker)
5. Make changes in _core code a repeat #3
6. [HA] bump versions in config.yaml and Changelog
7. PR in _core and in application

Proposed general way of working: each « application » is responsible for its compatibility with the _core code.
If an update of the _core code breaks the application, a issue fixing comit is possible in the _core.