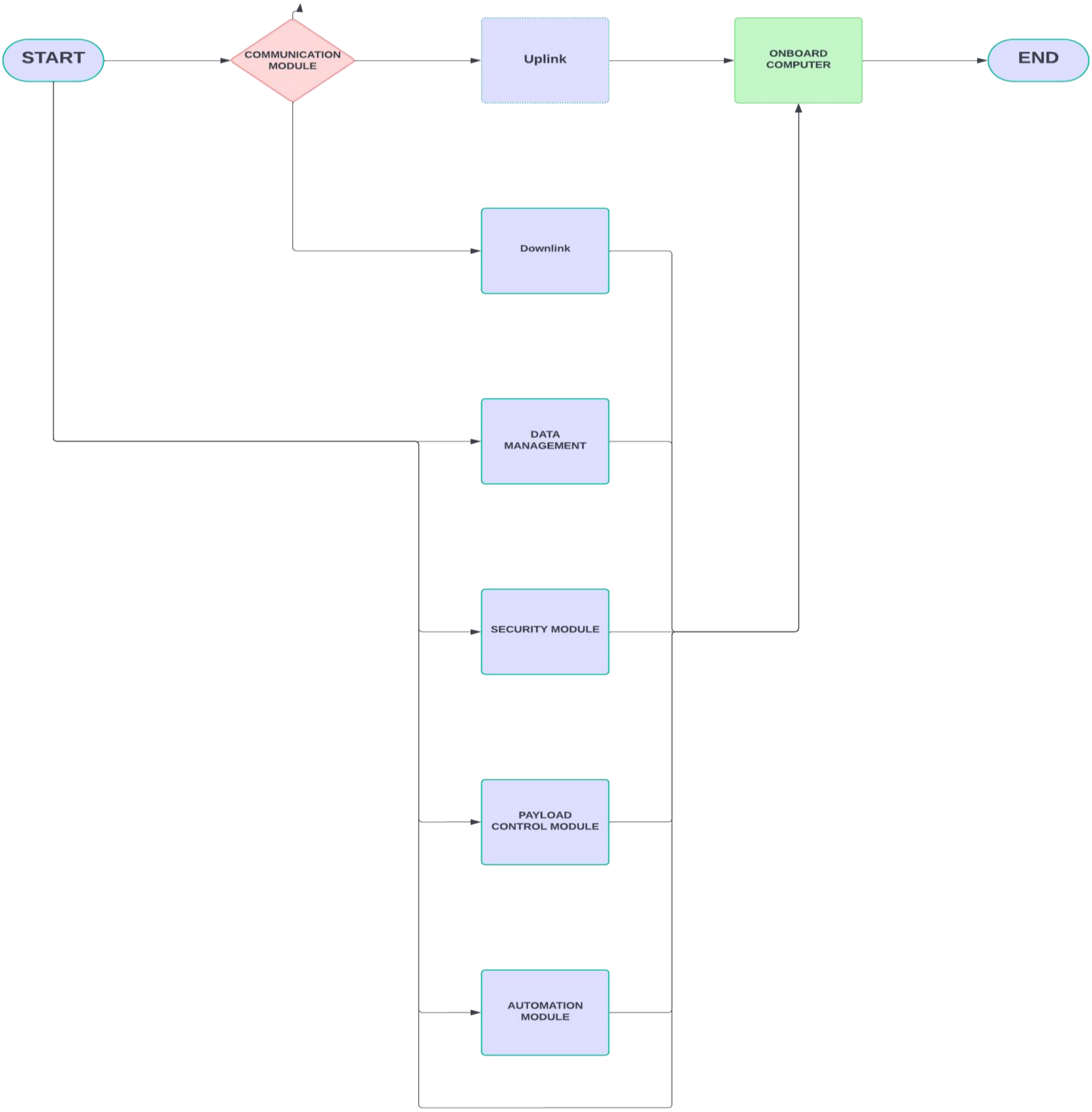
CubeSat for Rwandan Agricultural Monitoring (C-RAM) Mission Design



1. Operating System (OS)

a) **Real-time Operating System (RTOS):** Essential for managing the CubeSat's hardware and software resources in a timely and predictable manner.

    a) **Examples:** FreeRTOS (open-source), RTEMS (open-source), VxWorks (commercial)

2. Communication Software

a) **Communication Protocols:** Implements standards for space data communication, ensuring reliable data transfer between the CubeSat and ground station.

    I. **Example:** CCSDS (Consultative Committee for Space Data Systems) protocol suite.

b) **Data Handling:** Software to encode, decode, packetize, and manage data transmission.

    I. **Examples:** Custom software developed for the specific mission, libraries for CCSDS protocols.

3. Payload Control Software

a) **Sensor Drivers:** Software that allows the onboard computer to communicate with and control the sensors (camera, weather sensors, etc.).

    I. **Examples:** Often custom-written for specific sensors, but can sometimes use existing libraries or drivers.

b) **Image Processing Libraries:** For processing image data from cameras (e.g., calibration, noise reduction, feature extraction).

    I. **Examples:** OpenCV (open-source), GDAL (open-source)

4. Data Management Software

a) **File System:** Organizes and manages data storage on the CubeSat's memory.

    I. **Examples:** FATFS (open-source), LittleFS (open-source)

b) **Database Management:** May be used for efficient storage and retrieval of large datasets.

    I. **Example:** SQLite (open-source)

5. Automation Software

a) **Scripting Languages:** Used for automating tasks and creating scripts for specific operations.

    I. **Examples:** Python, Lua

b) **Scheduling Tools:** For scheduling tasks like data acquisition and downlinking.

   I. **Examples:** Custom schedulers, cron-like utilities for embedded systems.

6. Security Software

a) **Encryption Libraries:** To encrypt communication channels and protect sensitive data.

   I. **Examples:** WolfSSL (open-source), mbed TLS (open-source)

b) **Authentication Protocols:** To verify the identity of the ground station and prevent unauthorized access.

Development Tools

a) **Integrated Development Environment (IDE):** For writing, compiling, and debugging the CubeSat software.

   I. **Examples:** Eclipse, Visual Studio Code

b) **Software Version Control:** To track changes to the code and manage different versions.

   I. **Example:** Git

Factors to Consider When Choosing Software

a) **Resource Constraints:** CubeSats have limited processing power, memory, and storage, so software must be lightweight and efficient.

b) **Reliability:** Software must be robust and reliable to ensure the success of the mission.

c) **Power Consumption:** Software should be optimized to minimize power usage.

d) **Open Source vs. Commercial:** Open-source software can be cost-effective, but commercial options may offer better support and features.

This list provides a general overview of the software needed for a CubeSat mission for agriculture. The specific software requirements will vary depending on the mission objectives, payload, and chosen hardware.